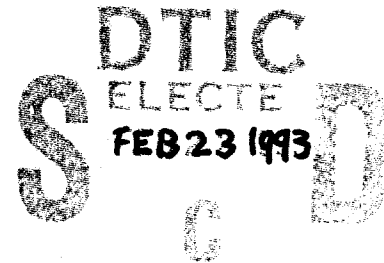


AD-A261 302

NISTIR 4329

**GRAPHICS
STANDARDS IN THE
COMPUTER-AIDED
ACQUISITION AND
LOGISTIC SUPPORT
(CALS) PROGRAM,
FISCAL YEAR 1989
VOLUME 1: TEST
REQUIREMENTS
DOCUMENT,
EXTENDED CGM
(CGEM)**



**Daniel R. Benigni
Editor**

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
National Computer Systems Laboratory
Gaithersburg, MD 20899

May 1990

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and For Special
A-1	

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited



93 2 22 022

93-03667



U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director

NISTIR 4329

**GRAPHICS
STANDARDS IN THE
COMPUTER-AIDED
ACQUISITION AND
LOGISTIC SUPPORT
(CALS) PROGRAM,
FISCAL YEAR 1989
VOLUME 1: TEST
REQUIREMENTS
DOCUMENT,
EXTENDED CGM
(CGEM)**

**Daniel R. Benigni
Editor**

**U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
National Computer Systems Laboratory
Gaithersburg, MD 20899**

May 1990

Accession For	
1. Title	2. Author
3. Subject	4. Date
5. Source	6. Price
7. Availability	8. Remarks
Accession Number	
Doc	Accession Number
A-1	



**U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director**

EXECUTIVE SUMMARY

The overall objective of the Department of Defense Computer-aided Acquisition and Logistic Support (CALS) Program is to integrate the design, manufacturing, and logistic functions through the efficient application of computer technology. NIST has been funded since Spring 1986 to recommend a suite of industry standards for system integration and digital data transfer, and to accelerate their implementation.

During FY86 NIST tasks for CALS in the area of graphics standards focused on identification of recommended standards to OSD which would be applicable to the DoD environment; comparison of graphics standards among themselves and with product data standards; and the status of ongoing graphics standards efforts as well as related validation efforts¹. CALS' needs for graphics standards were assessed, and an architecture for their inclusion into specific CALS programs was created. Finally, a plan was recommended for accelerating the development, related validation efforts and implementation of graphics standards into the CALS program.

Building on the knowledge and experience gained during FY86, NIST tasks for CALS in the area of graphics standards in FY87 emphasized the particular graphics standard dealing with the transfer of pictorial data from one system to another, namely the Computer Graphics Metafile (CGM) standard (FIPS PUB 128)². Graphics tasks included an assessment of raster-to-vector conversion technology, and where the CGM might fit into that process; efforts toward development of CGM validation routines; injection of CALS requirements into the Extended CGM (CGEM) standards work, as well as into the CGM Registration process. In addition, functional requirements and conceptual design documents were completed for a reference implementation for CGM; a design specification was created for an IGES-to-CGM translator; and a preliminary CALS Application Profile for CGM was formed from the application profile work of the MAP/TOP organization.

During FY88, NIST tasks for CALS in the graphics standards area were in large measure a continuation of those efforts begun the

¹Kemmerer, S., Editor, "Final NBS Report for CALS, FY86," U.S. Department of Commerce, National Bureau of Standards, NBSIR 87-3566, May 1987.

²Kemmerer, S., Editor, "A Collection of Technical Studies Completed for the Computer-aided Acquisition and Logistic Support (CALS) Program, Fiscal Year 1987," U.S. Department of Commerce, National Bureau of Standards, NBSIR 88-3727, March 1988.

year before³. Final text was completed for the initial publication of a Military Specification which is the CALS Application Profile for CGM. NIST continued to participate in graphics standards work in support of CALS requirements in the areas of CGM conformance testing, the CGEM, and CGM Registration. In the particular area of CGM conformance testing, the needs for testing both to FIPS 128 and to the Application Profile for CGM were identified; existing commercial implementations of CGM were analyzed and compared functionally to both CGM and Application Profile requirements; and required CGM conformance testing tasks were described in detail, responsibilities in the testing process delineated, and the impact of CGM testing was assessed both for CALS and the commercial marketplace.

This collection of reports represents the continuing efforts of the Graphics Software Group of NIST/NCSL in FY89 in support of computer graphics standards for CALS, and in particular CGM⁴. It provides a progress report on continuing graphics standards efforts related to the Computer Graphics Metafile (CGM) standard, including the Extended CGM (CGEM), Graphics Registration, and the CGM Application Profile for CALS (or MIL-D-28003). In addition, the creation of a Test Requirements Document for MIL-D-28003 is detailed. This Test Requirements Document will provide the basis for developing conformance tests to determine compliance with MIL-D-28003.

This report is subdivided into four separate final CALS deliverables, entitled as follows:

1. Test Requirements Document for CALS CGM Conforming Basic Metafiles
2. Injection of CALS Requirements in the Extended CGM (CGEM) Standards Work
3. MIL-D-28003 Revision Recommendations
4. CGM Registration in Support of CALS Requirements

³Morgan, Roy S., Editor, "A Collection of Technical Studies Completed for the Computer-aided Acquisition and Logistic Support (CALS) Program, Fiscal Year 1988, U.S. Department of Commerce, National Institute of Standards and Technology, NISTIR 4315, 4316, and 4317.

⁴The publishing of this collection of reports does not imply that the CALS Office has endorsed the conclusions or recommendations presented.

An additional deliverable completed for CALS by the Graphics Software Group during FY89 detailed the impact that two other graphics standards (namely PHIGS, or the Programmers Hierarchical Interactive Graphics System, and PIK, or the Programming Imaging Kernel)⁵ will have on the CALS environment. It was published under separate cover, and is available through the CALS Policy Office or through the National Technical Information Service.

⁵Kemmerer, Sharon J., and Skall, Mark W., "Graphics Application Programmer's Interface Standards and CALS," U.S. Department of Commerce, National Institute of Standards and Technology, NISTIR 89-4199, October 1989.

CONTRIBUTORS

NIST would like to acknowledge the major technical contributors to the separate reports contained herein. They are:

Peter R. Bono Associates, Inc., in particular Dr. Peter R. Bono, for his work on (1) above;

Henderson Software Company, in particular Mr. Lofton Henderson, for his work on (2) and (3) above, and

GSC Associates Inc., in particular Dr. George S. Carson, for his work on (4) above.

The editor would also like to gratefully acknowledge the efforts of those who participated in the review process of the documents presented in this report, namely:

David K. Jefferson
Sharon J. Kemmerer
Roy S. Morgan
Susan Quinn Sherrick
Lynne Rosenthal
Mark W. Skall

FINAL REPORT

CALS FY89 SOW TASK 4.1.1

TEST REQUIREMENTS DOCUMENT FOR
CALS CGM CONFORMING BASIC METAFILES

TABLE OF CONTENTS

I.	OVERVIEW AND METHODOLOGY	1
A.	Overview	1
B.	Purpose	1
C.	Scope	1
D.	Use	1
E.	Methodology	2
II.	REFERENCE ARCHITECTURE FOR A CGM PARSER/VERIFIER	3
A.	Overview	3
B.	Parser/Verifier Operation	3
1.	Step 1: Physical to Logical Mapping	3
2.	Step 2: Lexical Phase	4
3.	Step 3: Element Recognition	4
4.	Step 4: Element Processing	4
5.	Step 5: Special Element Processing	6
6.	Step 6: Final Processing	9
7.	Step 7: Reporting	10
III.	REQUIREMENTS EXTRACTED FROM ISO 8632-1:1987	11
A.	Explanation of Contents	11
B.	Organization of Requirements	11
1.	Required Elements	12
2.	Required Order of Elements	12
3.	Some Global Constraints on the Content of the CGM	14
4.	Metafile Defaults Replacement Requirements	16
5.	Non-final Text Requirements	17
6.	String Contents Requirements	19
7.	General Assertions about Parameter Values	20
8.	Specific Parameter Range Constraints	21
IV.	REQUIREMENTS EXTRACTED FROM ISO 8632-3:1987	29
A.	Explanation of Contents	29
B.	Organization of Requirements	29
C.	Errors in the CGM Specification	29
V.	REQUIREMENTS EXTRACTED FROM MIL-D-28003	39
A.	Explanation of Contents	39
B.	Organization of Requirements	39
1.	General CALS	40
2.	Physical to Logical Mapping	40
3.	Metafile Structure and Ordering	41
4.	Parameter Decoding	42
5.	Parameter Ranges	42
6.	Size Constraints	47
7.	GDP and ESCAPE Elements	47
8.	Colour/Pattern Usage Rules	48
VI.	CROSS REFERENCE TABLE BETWEEN REQUIREMENTS AND TESTING	51

TABLE OF CONTENTS (Continued)

APPENDIX A:	CGM ELEMENT ORDERING REQUIREMENTS	61
APPENDIX B:	CGM ELEMENT LENGTH REQUIREMENTS FROM ISO 8632-3:1987	71
APPENDIX C:	ALLOWABLE PARAMETER RANGES FOR BINARY ENCODED CGM ELEMENTS	79

ABSTRACT

The purpose of this report is to satisfy the requirements of CALS FY89 Statement of Work Task 4.1.1, which states:

Continue to accelerate the development of CGM validation routines and ensure the input of CALS requirements.

In previous related tasks, NIST/NCSL has: developed a plan for the development of conformance tests for both FIPS PUB 128 (CGM) and MIL-D-28003; compared the variability of commercial CGM generator implementations and how well they conform to MIL-D-28003; and assessed the impact of CGM test development strategy with respect to the CALS environment and the marketplace. Finally, as part of last fiscal year's work on CALS, NIST/NCSL prepared a comprehensive list of tasks that must be accomplished in order to put into place a testing service for both FIPS PUB 128 and MIL-D-28003. This report is a follow-on to that work, and fulfills one of the most important tasks identified concerning the development of a Test Method, i.e. to create a Test Requirements Document.

I. OVERVIEW AND METHODOLOGY

A. Overview

The Computer Graphics Metafile (CGM) standard (ISO 8632; ANSI/X3.122; FIPS 128) is a data transfer standard. In particular, it specifies the content of a logical file to be used in storing and interchanging picture descriptions among applications. Conformance statements in the CGM standard apply to instances of CGMs and not to generators (writers) of CGMs nor to interpreters (readers) of CGMs. Consequently, the standard specifies mainly syntactic requirements; there are very few semantic requirements to be met by a conforming instance of a CGM.

B. Purpose

The purpose of a Test Requirements document is to gather together all requirements that must be satisfied in order for a given instance of an implementation of a graphics standard to be in conformance with that standard. In the case of the CGM standard, an instance of a CGM whose conformance is in question will be called a CGM-under-test.

C. Scope

This test requirements document is meant to apply to CGMs that attempt to comply with MIL-D-28003, the CALS Application Profile for the CGM. Consequently, the document has gathered only those requirements that apply to Binary Encoded CGMs; i.e., those that conform to Parts 1 and 3 of the CGM standard.

D. Use

This document is intended for use by (1) programmers who must build a testing tool capable of determining whether a given instance of a CGM is in conformance with MIL-D-28003 and (2) buyers who need to judge, by whatever means are available, whether a given CGM is in conformance with MIL-D-28003. In addition, programmers responsible for producing products with capabilities for CGM generation or interpretation will find this document valuable.

Although specifically targeted at CALS-conforming CGMs, the document presents the CALS-specific requirements of MIL-D-28003 separately, so that the document is of value to anyone needing to test Binary Encoded CGMs for conformance to the CGM standard. In addition, because the Functional Requirements are documented separately from the Encoding Requirements, this document could also serve as the basis for a more comprehensive CGM Test Requirements document that deals with all three standardized encodings of the CGM.

E. Methodology

The CGM Functional Description, ISO 8632-1:1987, was systematically read and all requirements relating to the conformance of instances of CGMs were extracted. Section III presents the results of that analysis. A similar approach was taken for the CGM Binary Encoding, ISO 8632-3:1987, and the CALS Application Profile, MIL-D-28003. The results are summarized in Sections IV and V, respectively. In these sections, like requirements are grouped. [NOTE: Because this document is intended to be used in the development of an international CGM conformance testing tool, the requirements were derived from ISO 8632:1987. It should be further noted that ISO 8632 and ANSI X3.122 are identical except for document layout and style (i.e., page numbers may differ).]

From the total set of requirements, three major summary tables were assembled and are presented as Appendices. First, Appendix A contains the description of a finite state machine that enforces exactly all CGM Element ordering requirements contained in the standard. Next, Appendix B contains, in tabular form, the parameter list lengths for all elements in a CALS-conforming CGM. Finally, Appendix C presents a matrix stating the requirements that must be met by each parameter of each CGM Element appearing in a CALS-conforming CGM.

These tables summarize and account for over two-thirds of the specific requirements statements found in the CGM standard. The conformance implications of the remaining requirements statements cannot fruitfully be characterized in a simple table. Instead, an expository technique has been chosen, which is elaborated in Section II, based on describing the behavior of a hypothetical "black box," whose task it is to determine whether a given CGM-under-test is in conformance with MIL-D-28003.

The black box is called a "CGM Parser/Verifier." Its operation is broken down into six major steps, which can be thought of as relatively autonomous program modules, plus a reporting step. Section II gives a complete description of the behavior of the parser/verifier and assumes that the parser/verifier uses the information embodied in the summary tables of Appendices A, B, and C.

To complete the report, Section VI gives a cross reference between each raw requirement statement and the Table (Appendix A, B, or C) or Step (in Section II) where the test requirement is represented for testing.

II. REFERENCE ARCHITECTURE FOR A CGM PARSER/VERIFIER

A. Overview

The CGM standard, ISO 8632:1987, Parts 1 (Functional Description) and 3 (Binary Encoding), specifies the content and structure of a conforming metafile. This section describes the high-level design of a CGM parser/verifier whose objective is to determine whether a given CGM is conforming. If the CGM-under-test is conforming, the parser/verifier should produce a conformance report stating this fact. If the CGM-under-test does not conform to the standard, the conformance report should include a description of each conformance violation discovered including its location (as a logical octet offset from the start of the CGM).

MIL-D-28003, "Digital Representation for Communication of Illustration Data: CGM Application Profile," places additional constraints on the structure and content of a CALS basic conforming metafile. In the following description of the parser/verifier operation, CALS-specific tests are separated into their own paragraphs and preceded with the phrase "CALS".

B. Parser/Verifier Operation

The parser/verifier is divided into a number of relatively autonomous modules described in the sections that follow.

1. Step 1: Physical to Logical Mapping

The CGM-under-test is opened and the initial information is read from the file into an internal buffer. Because the CGM is logically viewed as a continuous stream of bits, organized as octets and words, according to specific definitions in the standard, it is in this module that all operating-system and programming-language-specific matters like byte swapping and fixed-length and variable-length record structures are dealt with. For the rest of this discussion, it is assumed that all subsequent steps are able to get any number of octets from the logical CGM-under-test in the correct order without worrying about whether there are file markers, record markers, and the like embedded in the stream of octets returned to the lexical phase (step 2).

CALS. MIL-D-28003 specifies that basic conforming metafiles consist of 80-octet fixed-length records. If transmitted on magnetic tape in accordance with MIL-STD 1840A, they are blocked into 800-octet physical records (i.e., 10 records per block).

2. Step 2: Lexical Phase

Two octets are obtained from the CGM-under-test. They represent the command header (short form). The element class and id are extracted and saved. The length is examined to determine if a long-form element is present. If so, another two octets are obtained. The parameter length information (short or long) is saved. Bit 15 is examined in the long form to determine whether data partitioning is in effect. The starting point of the next command header (or next data partition) is calculated, taking into consideration that command headers and partitions start on word boundaries.

One special piece of processing that can occur here is detecting when all the elements comprising the parameter data in a METAFILE DEFAULTS REPLACEMENT element have been picked up. Prior to getting the command header of the first element following the METAFILE DEFAULTS REPLACEMENT element, the state should be set to MDOP.

CALS. In a basic conforming metafile, the METAFILE DEFAULTS REPLACEMENT element shall not be partitioned. If partitioning is encountered, generate an appropriate CALS-conformance violation message, write it to the profile error report file, and increment the count of such errors.

3. Step 3: Element Recognition

If the class/id is recognized as one of the valid opcodes, a frequency count for each opcode encountered is incremented. Then, processing passes to Step 4 (Element Processing).

Otherwise, a conformance violation message is formulated and output to an error report file, and the error count for this category of error is incremented. Then the octet pointer into the CGM-under-test is positioned to the start of the next command header, skipping past all parameter data for the illegal element, including any data partitions that might be encountered. Control is returned to step 2.

4. Step 4: Element Processing

If this element is the first element in the metafile and it is not the BEGIN METAFILE element, the parser/verifier immediately halts processing with an appropriate message.

Each valid CGM element needs its own processing section, because different specific actions need to be taken for each element. However, the processing follows a familiar pattern, which is described in the following paragraphs.

It is assumed that the parser/verifier has established a set of global variables which contain both the current metafile default values for all Metafile Descriptor, Picture Descriptor, Control, and Attribute elements and the current parser/verifier values for all Picture Descriptor, Control, and Attribute elements. It is also assumed that the parser/verifier is checking element ordering rules by implementing the finite-state machine logic according to the description provided in Appendix A.

(Step 4A: Check Element Order) First, the current state of the parser/verifier is compared against the states allowed for this element. If the element is not allowed in this state, an appropriate error message is formulated and written and the error count updated for this class of error. However, processing is allowed to continue.

CALS. MIL-D-28003 places some constraints on the use of ESCAPE elements, and GDP elements are not permitted. These constraints are shown in the state table summary section (Appendix A).

(Step 4B: Acquire Parameter) The correct number of octets for each parameter is picked up in turn from the CGM-under-test. This number often varies according to the current variable settings. If the number of octets needed would exceed the number of octets remaining available for this element, processing of this element is aborted and control passes to Step 2, after an appropriate error message is formulated and written and the error count updated for this class of error. During this step, one must be careful to observe rules relating to byte and word alignment, where they apply (e.g, in the CELL ARRAY element)--see Section IV, requirements B2, B13, B14, B15, B18, B19, B32, and B33.

CALS. MIL-D-28003 restricts the range of allowed precisions to a subset of all those allowed by the CGM standard. The length (in octets) of each element in a basic conforming metafile is shown in a summary table (Appendix B). All valid element lengths are shown where the basic set permits more than one precision.

(Step 4C: Decode Parameter) The parameter is decoded according to the data type expected for this element. If any error occurs upon decoding, an appropriate error message is formulated and written and the error count updated for this class of error.

(Step 4D: Check Parameter Range) The parameter is then checked against any range constraints that might apply. These constraints might be universal (e.g., an enumerated type must be either 0 or 1) or might be dependent upon the current variable settings (e.g., all colour indices must be non-negative and no greater than the maximum colour index). If any parameter fails a range check, an appropriate error message is formulated and written and the error count updated for this class of error. Range checks for strings include verifying that the string length is non-negative and that the strings contain only legal character codes. This latter check may depend upon the value of the Character Coding Announcer.

CALS. MIL-D-28003 permits only a subset of the permissible CGM values for each parameter to be present in a basic conforming metafile. Appendix C shows the allowable range for each parameter and is annotated with these additional CALS-specific range constraints, when the CALS Basic Set is a proper subset of the permissible CGM values according to the CGM standard.

(Step 4E: Perform Element-specific Special Processing) Once all the parameters for an element have been acquired, any special processing for that element can take place. Special processing for each element or group of elements is described in Step 5.

5. Step 5: Special Element Processing

BEGIN METAFILE: Set all current metafile defaults to the fixed defaults specified in the standard. Save the string parameter contents. Set state to MDOP.

END METAFILE: Set state to MFCL. Go to Step 6 (Final Processing).

BEGIN PICTURE: Reset all current picture descriptor variables to the corresponding values for the current metafile defaults. Save the contents of the string parameter of each picture. Set state to PDOP. At the first BEGIN PICTURE, examine the frequency count data to verify that the METAFILE VERSION and METAFILE ELEMENT LIST elements were present in the Metafile Descriptor.

BEGIN PICTURE BODY: Set state to PBOP.

END PICTURE: Set state to PICL.

CALS. Perform the Colour/Pattern Usage checks described in the following:

The colour index used/set/redefined information and the pattern index used/set/redefined information shall be compared with the requirements of MIL-D-28003. CALS conformance violation messages shall be generated, reported, and counted, under the following circumstances:

- (a) Not all colour indexes used were set, unless none of the indexes were set.
- (b) The colour redefinition list is non-empty.
- (c) The pattern redefinition list is non-empty.

Metafile Descriptor Elements (general): Set the corresponding value in the current metafile defaults global data structure.

METAFILE DESCRIPTION: Save the contents of the string parameter.

CALS. Verify that the description contains "MIL-D-28003/BASIC-1" as a substring. Verify that some additional text is present-text that could serve to identify the company or product.

METAFILE ELEMENTS LIST: Mark the element frequency count global data structure with the opcodes in the list. Correctly expand the codes for "drawing set" and "drawing plus controls set".

METAFILE DEFAULTS REPLACEMENT: Set state to MMDR. Set octet counter (file offset) to correct value so that state can be set back to MDOP when all the parameters for this element have been processed (see Step 2 discussion).

CALS. FONT LIST: All the font names encountered in the font list must match one of the sixteen Hershey font names specified in MIL-D-28003.

Picture Descriptor Elements (general): Set the corresponding value in the current variables global data structure if the state is PDOP and in the current metafile defaults global data structure if the state is MMDR.

CALS. BACKGROUND COLOUR. Follow the same processing as required for colour table elements.

Control Elements (general): Set the corresponding value in the current variables global data structure if the state is PBOP and in the current metafile defaults global data structure if the state is MMDR.

Geometric Primitives (general): Check for any special constraints on the number of entities expected. For example, check for at least 2-point polylines, 3-point polygons, and disjoint polylines with an even number of points.

CALS. GDP: Report an error, because no GDPs are registered as yet.

CALS. The number of colour values shall not exceed 1,048,576 in a CELL ARRAY element. The number of points in any metafile element shall not exceed 1024. No string parameter, with the exception of data records, shall exceed 254 characters in length; data records shall not exceed 32767 characters.

CALS. If the default colour index for this type of primitive is the current colour index, mark the corresponding internal colour table entry as used. If the entry is not already marked as set, then generate, record, and count a CALS Application Profile conformance violation if any other index has already been set.

CALS. If this is a fill-area type primitive, if the current interior style is "pattern," and if the default pattern index is the current pattern index, mark the corresponding internal pattern table entry as used. If the entry is not already marked as set, then generate, record, and count a CALS Application Profile conformance advisory (which may become a violation in a future version of MIL-D-28003).

"not final" TEXT and RESTRICTED TEXT: Set state to TXOP.

"final" APPEND TEXT: Set state to PBOP.

Attribute Elements (general): Set the corresponding value in the current variables global data structure, if the state is PBOP, and in the current metafile defaults global data structure, if the state is MMDR.

CALS. The number of colour values shall not exceed 2048 in a pattern table and 256 in a COLOUR TABLE element. No string parameter, with the exception of data records, shall exceed 254 characters in length; data records shall not exceed 32767 characters.

CALS. Colour Value Selection Elements (for indexed colour): These elements include LINE COLOUR, MARKER COLOUR, TEXT COLOUR, FILL COLOUR, and EDGE COLOUR. Also included are the colour index aspects of the corresponding bundles, when the appropriate aspect source flag controlling the setting of colour is individual. Mark the corresponding internal colour table entry as used. If the entry is not already marked as set, record and count a CALS Application Profile conformance violation if any other index has

already been set. This processing does not take place if the parser/verifier is in the MMDR state (that is, if default colour indexes are being specified).

CALS. PATTERN INDEX: Mark the corresponding internal pattern table entry as used. If the entry is not already marked as set, generate, record, and count a CALS Application Profile conformance advisory (which may become a violation in a future version of MIL-D-28003). This processing does not take place if the parser/verifier is in the MMDR state (that is, if a default pattern index is being specified).

CALS. PATTERN TABLE: Save the pattern in the internal pattern table. Mark the entry as "set." If the entry is already marked as used and the pattern table parameter values are different from the values already set in the internal table, record this pattern index as "redefined". This processing does not take place if the parser/verifier is in the MMDR state (that is, if a default pattern table is being specified). Note also that no redefinition of pattern table entries is allowed once the first primitive has been encountered by the parser.

CALS. COLOUR TABLE: Save the colour values in the internal colour table. Mark the entry as "set." If the entry is already marked as used and the colour table parameter values are different from the values already set in the internal table, record this colour index as "redefined". This processing does not take place if the parser/verifier is in the MMDR state (that is, if a default colour table is being specified). Note also that no redefinition of colour table entries is allowed once the first primitive has been encountered by the parser.

CALS. ESCAPE: Report an error if the id is not -301, -302, or -303, because these are the only ESCAPES authorized for use for CALS.

6. Step 6: Final Processing

Once the entire CGM-under-test has been interpreted, there are still two global checks that remain to be accomplished.

(Step 6A: Required Elements) The frequency count data is examined to verify that END METAFILE has occurred.

(Step 6B: Correctness of METAFILE ELEMENT LIST) The frequency count data is compared with the elements marked as a result of their presence in the METAFILE ELEMENT LIST to verify that every element actually present in the metafile was mentioned in the METAFILE ELEMENT LIST.

7. Step 7: Reporting

A FIPS PUB 128 conformance report is produced. The report should contain, at a minimum:

- (a) The file name of the CGM-under-test.
- (b) The (logical) size of the file in octets.
- (c) The contents of the string parameters associated with BEGIN METAFILE and METAFILE DESCRIPTION (if present).
- (d) A count of the number of pictures present and the starting octet count and content of the string parameter associated with each BEGIN PICTURE.
- (e) A statement of conformance reporting the total number of elements tested and the number and type of errors found (if any).
- (f) Specific error messages for each conformance violation detected along with information that permits localization of the error (e.g., element name and offset into the file).

CALS. A MIL-D-28003 conformance report supplement should be appended to the basic conformance report. It should provide:

- (a) A statement of CALS conformance reporting the number and type of Application Profile errors found (if any).
- (b) Specific error messages for each CALS CGM Application Profile conformance violation detected along with information that permits localization of the error (e.g., element name and offset into the file).

III. REQUIREMENTS EXTRACTED FROM ISO 8632-1:1987

A. Explanation of Contents

The CGM Functional Description, ISO 8632-1:1987, was systematically read and all requirements relating to the conformance of instances of CGMs were extracted. Each separate statement of requirements is assigned a number and quoted. The requirement numbers are assigned sequentially from one and all start with the letter "F," indicating that these requirements come from the CGM Functional Description, Part 1. A suffix--"a," "b," "c"--is added if the identical requirement is stated multiple times in different places.

A citation for each statement (shown in boldface) is given according to the following scheme: The Part 1 clause number is provided on the first line; on the second line, for Clause 4 and Clause 6 citations, the paragraph number (p) and sentence number(s) are specified as "#p/s", followed by the page number on which the text occurs. For Clause 5 citations, the only difference is that the second line starts with either a "P" or "D", rather than a "#". "P" means that the information is contained in the Parameters section of the clause and "D" that the information is contained in the Description section of the clause.

Any commentary on the requirements statement is shown as a note enclosed in brackets ([NOTE:...]).

B. Organization of Requirements

The requirements are grouped into eight main categories. Within each category, the requirements are generally stated in order of their clause number. The only exception to this rule is when a requirement is stated in several places in the standard. These requirement statements are grouped together.

The eight categories of requirements statements are:

- Required Elements
- Required Order of Elements
- Some Global Constraints on the Content of the CGM
- Metafile Defaults Replacement Requirements
- Non-final Text Requirements
- String Contents Requirements
- General Assertions about Parameter Values
- Specific Parameter Range Constraints

1. Required Elements

F1: Clause 4.1
#4/1 p.9

A minimal correct metafile consists of BEGIN METAFILE, a Metafile Descriptor consisting of METAFILE VERSION and METAFILE ELEMENT LIST, and END METAFILE.

F2: Clause 5.2.1
D1/3 p.44

BEGIN METAFILE shall occur exactly once in a metafile.

F3: Clause 5.3.1
D1/2 p.47

[The METAFILE VERSION] element shall occur in the Metafile Descriptor of every metafile.

F4: Clause 5.3.11
D1/3 p.50

METAFILE ELEMENT LIST shall occur in the Metafile Descriptor of every metafile.

2. Required Order of Elements

F5: Clause 4.2
#1/1 p.9

Every metafile starts with a BEGIN METAFILE element ...

F6: Clause 5.2.1
D1/1 p.44

[BEGIN METAFILE] is the first element of a metafile.

F7: Clause 4.2
#1/1 p.9

Every metafile ... ends with an END METAFILE element.

F8: Clause 5.2.2
D1/1 p.44

[END METAFILE] is the last element of a metafile.

F9: Clause 4.3
#2/3 p.10

The CGM contains a single Metafile Descriptor ... [which] immediately follows the BEGIN METAFILE element in a metafile (with the possible exception of intervening external and escape elements).

[NOTE: This implies that Metafile Descriptor elements may appear only before the first BEGIN PICTURE element in the metafile, if the metafile contains any pictures.]

F10: Clause 4.3
#2/4 p.10

External and escape elements may appear anywhere between the BEGIN METAFILE element and first BEGIN PICTURE element (if one exists) and after the last END PICTURE element (if one exists) and the END METAFILE element.

F11: Clause 4.4
#1/3 p.12

If included in a picture, [Picture Descriptor] elements shall appear after the BEGIN PICTURE element and before the BEGIN PICTURE BODY element.

F12: Clause 5.4.1
D1/5 p.56

If used, SCALING MODE shall appear in the Picture Descriptor.

F13: Clause 5.4.2
D2/4 p.56

If used, COLOUR SELECTION MODE shall appear in the Picture Descriptor.

F14: Clause 5.4.3
D2/3 p.57

If used, LINE WIDTH SPECIFICATION MODE shall appear in the Picture Descriptor.

F15: Clause 5.4.4
D2/3 p.57

If used, MARKER SIZE SPECIFICATION MODE shall appear in the Picture Descriptor.

F16: Clause 5.4.5
D2/3 p.57

If used, EDGE WIDTH SPECIFICATION MODE shall appear in the Picture Descriptor.

F17: Clause 4.4
#1/4 p.12

Escape and external elements are permitted in the Picture Descriptor.

F18: Clause 4.5
#1/2 p.14

Control elements ... may appear in the picture bodies in the metafile.

F19: Clause 4.9
#1/1 p.39

External elements ... may appear anywhere in the CGM.

F20: Clause 4.10
#Fig. 12 p.41

Control, Graphical Primitive, and Attribute elements can appear only while a picture is "open," that is, between a BEGIN PICTURE BODY element and the next END PICTURE element.

F21: Clause 5.2.5
D2/1 p.46

Only external and escape elements may occur between END PICTURE and BEGIN PICTURE or between END PICTURE and END METAFILE.

3. Some Global Constraints on the Content of the CGM

F22: Clause 4.3
#2/1 p.10

The METAFILE ELEMENT LIST lists at least those standardized elements that occur in the metafile.

F23: Clause 5.3.11
D1/1 p.50

All of the elements that may be encountered in the metafile and that are not mandatory are listed [in the METAFILE ELEMENT LIST].

F24: Clause 4.7.7
#5/2 p.38

There is a Metafile Descriptor element, COLOUR VALUE EXTENT, which allows metafile generators to specify the minimum and maximum metafile colour values.

[NOTE: This requirement implies that all colour values should be checked to ensure that they lie in the range specified in the COLOUR VALUE EXTENT.]

F25: Clause 5.3.10
D1/1-2 p.49

[For COLOUR VALUE EXTENT] the parameters represent an extent which bounds the direct colour values that will be encountered in the metafile. It need not represent the exact extent of colour values contained in the metafile.

F26: Clause 5.3.9
D1/1 p.49

[For MAXIMUM COLOUR INDEX] the parameter represents an upper bound (not necessarily the least upper bound) on colour index values that will be encountered in the metafile.

[NOTE: Therefore, the value of this parameter should be greater than or equal to all colour indices found in the metafile.]

F27: Clause 5.4.2
D2/1 p.56

Only one colour mode may be used within a picture.

F28: Clause 5.4.3
D2/1 p.57

Only one line width mode may be used within a picture.

F29: Clause 5.4.4
D2/1 p.57

Only one marker size mode may be used within a picture.

F30: Clause 5.4.5
D2/1 p.57

Only one edge width mode may be used within a picture.

4. Metafile Defaults Replacement Requirements

F31: Clause 4.4.4
#3/1 p.12

The default state of the [VDC] extent ... can be changed in the METAFILE DEFAULTS REPLACEMENT element in the Metafile Descriptor.

F32: Clause 4.4.5
#1/3 p.14

The default background colour [can be] specified in the METAFILE DEFAULTS REPLACEMENT element.

F33: Clause 5.3.12
P1/1 p.50

Picture Descriptor, Control, and Attribute elements [may appear in the METAFILE DEFAULTS REPLACEMENT element].

F34: Clause 5.3.12
D1/4 p.50

Any subset of the elements given defaults in clause 6 may be included [in the METAFILE DEFAULTS REPLACEMENT element].

F35: Clause 5.3.12
D2/3 p.50-1

An element [can] occur more than once in the default replacement list.

F36: Clause 6

#all p.101-103

By implication (all non-Metafile-Descriptor elements mentioned in Clause 6), any of the following elements may appear in a METAFILE ELEMENTS REPLACEMENT element:

SCALING MODE	CHARACTER EXPANSION FACTOR
COLOUR SELECTION MODE	CHARACTER SPACING
LINE WIDTH SPECIFICATION MODE	TEXT COLOUR
MARKER SIZE SPECIFICATION MODE	CHARACTER HEIGHT
EDGE WIDTH SPECIFICATION MODE	CHARACTER ORIENTATION
VDC EXTENT	TEXT PATH
BACKGROUND COLOUR	TEXT ALIGNMENT
VDC INTEGER PRECISION	CHARACTER SET INDEX
VDC REAL PRECISION	ALTERNATE CHARACTER SET INDEX
AUXILIARY COLOUR	FILL BUNDLE INDEX
TRANSPARENCY	INTERIOR STYLE
CLIP RECTANGLE	FILL COLOUR
CLIP INDICATOR	PATTERN INDEX
LINE BUNDLE INDEX	EDGE BUNDLE INDEX
LINE TYPE	EDGE TYPE
LINE WIDTH	EDGE WIDTH
LINE COLOUR	EDGE COLOUR
MARKER BUNDLE INDEX	EDGE VISIBILITY
MARKER TYPE	FILL REFERENCE POINT
MARKER SIZE	PATTERN TABLE
MARKER COLOUR	COLOUR TABLE
TEXT BUNDLE INDEX	ASPECT SOURCE FLAGS
TEXT FONT INDEX	
TEXT PRECISION	

5. Non-final Text Requirements

F37: Clause 4.6.3.2

#2/1 p.17

Changes to the text attributes TEXT FONT INDEX, CHARACTER EXPANSION FACTOR, CHARACTER SPACING, TEXT COLOUR, CHARACTER HEIGHT, CHARACTER SET INDEX, ALTERNATE CHARACTER SET INDEX, and TEXT BUNDLE INDEX, and to the control elements AUXILIARY COLOUR and TRANSPARENCY are permitted between a non-final text element and its succeeding APPEND TEXT element.

[NOTE: NB: Clause 4.7.6 and Figure 12 (p. 41) also indicates that TEXT PRECISION is okay, while the formal grammar (which is not part of the CGM standard) allows CHARACTER ORIENTATION, which is not mentioned in Clause 4.7.6 nor in Figure 12.]

F38: Clause 4.6.3.3
#1/4 p.17

The initial [text] element is always TEXT or RESTRICTED TEXT; subsequent elements may only be APPEND TEXT.

F39: Clause 4.7.6
#3/1 p.24

The attributes in the character representation and placement group [TEXT FONT INDEX, CHARACTER SET INDEX, ALTERNATE CHARACTER SET INDEX, TEXT PRECISION, CHARACTER EXPANSION FACTOR, CHARACTER SPACING, TEXT COLOUR, CHARACTER HEIGHT, AUXILIARY COLOUR, TRANSPARENCY] and TEXT BUNDLE INDEX may be changed within a string.

F40: Clause 4.10
#Fig. 12 p.41

No elements, other than the following list of elements, may appear between a "not final" TEXT or RESTRICTED TEXT element and a subsequent "final" APPEND TEXT element: "not final" APPEND TEXT, TEXT FONT INDEX, TEXT PRECISION, CHARACTER EXPANSION FACTOR, CHARACTER SPACING, TEXT COLOUR, CHARACTER HEIGHT, CHARACTER SET INDEX, ALTERNATE CHARACTER SET INDEX, TEXT BUNDLE INDEX, AUXILIARY COLOUR, TRANSPARENCY.

F41a: Clause 5.6.4
D3/1 p.63

also

F41b: Clause 5.6.5
D6/1 p.64

also

F41c: Clause 5.6.6
D3/1 p.65

The flag parameter is used to permit changing the following text attributes and control elements within a string which will be aligned as a single block: TEXT FONT INDEX, TEXT PRECISION, CHARACTER EXPANSION FACTOR, CHARACTER SPACING, TEXT COLOUR, CHARACTER HEIGHT, CHARACTER SET INDEX, ALTERNATE CHARACTER SET INDEX, TEXT BUNDLE INDEX, AUXILIARY COLOUR, and TRANSPARENCY.

F42a: Clause 5.6.4
D4/1-3 p.63

also

F42b: Clause 5.6.5
D7/1-3 p.64

also

F42c: Clause 5.6.6
D4/1-3 p.65

If the flag is set to 'not final', ... only the attribute setting elements listed above are allowed between this element and the APPEND TEXT element. With the exception of the ESCAPE element, no other metafile elements of any type are allowed.

6. String Contents Requirements

F43a: Clause 5.6.4
D1/3 p.63

also

F43b: Clause 5.6.5
D2/3 p.64

also

F43c: Clause 5.6.6
D1/4 p.65

Format effector control characters (such as CR, LF, BS, HT, VT, and FF) are permitted in a [TEXT] string but their interpretation is implementation-dependent.

F44a: Clause 5.6.4
D1/4 p.63

also

F44b: Clause 5.6.5
D2/4 p.64

also

F44c: Clause 5.6.6
D1/5 p.65

Control characters used for character set invocation and designation (SI, SO, ESC, SS2, and SS3) are permitted according to the setting of CHARACTER CODING ANNOUNCER.

F45: Clause 5.7.20
D3/1 p.89

If the appropriate CHARACTER CODING ANNOUNCER is selected, the SO and SI controls and ISO 2022 escape sequences may be embedded within the string parameters of text elements.

7. General Assertions about Parameter Values

F46: Clause 4.3.2
#1/3 p.10

Two shorthand names for CGM elements are also provided for use with the METAFILE ELEMENT LIST [element].

F47: Clause 4.3.2.1
#2/1 p.10

The drawing set includes a set of listed elements.

F48: Clause 4.3.2.2
#2/1 p.11

The drawing plus control set includes a set of listed elements.

F49: Clause 4.6.5
#2/1 p.19

The colour values [of a CELL ARRAY element] are either direct colour values or indexes into the COLOUR TABLE, according to the current COLOUR SELECTION MODE.

F50: Clause 5.4.2
D2/3 p.56

All occurrences of colour-setting elements (AUXILIARY COLOUR, LINE COLOUR, MARKER COLOUR, FILL COLOUR, EDGE COLOUR, TEXT COLOUR) as well as the colour lists of CELL ARRAY and PATTERN TABLE shall be in the current [colour selection] mode.

F51: Clause 4.6.5
#2/2 p.19

The colour values [of a CELL ARRAY element] are in the precision declared by a 'local colour precision' parameter of the CELL ARRAY element.

F52: Clause 5.4.6
D6/1 p.58

Specification of [VDC] values outside VDC EXTENT in parameters of CGM elements is permitted.

F53: Clause 5.6.9
D3/4-5 p.69

If the picture uses indexed colour selection, then the form of the [local colour precision CELL ARRAY] parameter is the same as that of COLOUR INDEX PRECISION. If the picture uses direct colour selection, then the form of the parameter is the same as that of COLOUR PRECISION.

F54: Clause 5.6.32
D3/4-5 p.96

If the picture uses indexed colour selection, then the form of the [local colour precision PATTERN TABLE] parameter is the same as that of COLOUR INDEX PRECISION. If the picture uses direct colour selection, then the form of the parameter is the same as that of COLOUR PRECISION.

8. Specific Parameter Range Constraints

F55: Clause 4.11
2/1-2 p.40

Applications therefore shall not use parameter values in the reserved ranges for implementation or private use. Those metafile elements that will be affected by registration of graphical items are: LINE TYPE, MARKER TYPE, HATCH STYLE, EDGE TYPE, FONT LIST, GENERALIZED DRAWING PRIMITIVE, ESCAPE.

F56: Clause 5.1
#11/4 p.43

Non-negative values [of type IX parameters] are reserved for (future) standardization.

F57: Clause 5.3.1
D2/1 p.47

This version of the CGM standard is version one (1).

[NOTE: This implies that the value of P1 must always be "1".]

F58: Clause 5.6.1
D1/1 p.62

[For POLYLINE] a line is drawn from ... the next-to-last point to the last point.

[NOTE: This might be taken to imply that the number of points must be at least 2.]

F59: Clause 5.6.2
D1/1 p.62

[For DISJOINT POLYLINE], a line is drawn from the starting point to the second point ...

[NOTE: This might be taken to imply that the number of points must be at least 2. Also, that there should be an even number of points in the point list.]

F60a: Clause 5.6.4
D5/3 p.63

also

F60b: Clause 5.6.5
D8/3 p.64

also

F60c: Clause 5.6.6
D5/2 p.65

Text elements with a null string parameter are legal.

F61: Clause 5.6.9
D1/1 p.69

In the general case, P, Q, and R [--the first three parameters of CELL ARRAY--] can delimit an arbitrary parallelogram.

[NOTE: This implies that the area specified by the vertices P, Q, and R should not be zero.]

F62: Clause 5.6.9
D4/1-2 p.69

Legal values of the 'local colour precision' include the legal values of COLOUR (INDEX) PRECISION. In addition, each encoding defines a special value, the 'default colour precision indicator', as an indicator that the colour specifiers of the [CELL ARRAY] element are to be encoded in the COLOUR (INDEX) PRECISION of the metafile; i.e., to indicate that the 'local colour precision' defaults to COLOUR (INDEX) PRECISION.

F63: Clause 5.6.10
D2/1 p.71

Non-negative values of the [GDP] identifier are reserved for registration and future standardization and negative values are available for private use.

[NOTE: This implies that 0 is not a legal GDP identifier at this time.]

F64: Clause 5.6.12
D2/1 p.72

Valid values of [a CIRCLE element's] radius are non-negative VDC.

F65: Clause 5.6.15
D5/1 p.75

Valid values of [a CIRCULAR ARC CENTRE element's] vector components are those which produce vectors of non-zero length.

F66: Clause 5.6.15
D6/1 p.75

Valid values of [a CIRCULAR ARC CENTRE element's] radius are non-negative VDC.

F67: Clause 5.6.16

D7/1 p.76

Valid values of [a CIRCULAR ARC CENTRE CLOSE element's] vector components are those which produce vectors of non-zero length.

F68: Clause 5.6.16

D8/1 p.76

Valid values of [a CIRCULAR ARC CENTRE CLOSE element's] radius are non-negative VDC.

F69: Clause 5.6.17

D3/1 p.76

Valid values of the three specifying points of the [ELLIPSE element] are those which yield three distinct points.

F70: Clause 5.6.18

D6/1 p.77

Valid values of the three specifying points of the [ELLIPTICAL ARC element] are those which yield three distinct points.

F71: Clause 5.6.18

D7/1 p.77

Valid values of [a ELLIPTICAL ARC element's] vector components are those which produce vectors of non-zero length.

F72: Clause 5.6.19

D7/1 p.78

Valid values of the three specifying points of the [ELLIPTICAL ARC CLOSE element] are those which yield three distinct points.

F73: Clause 5.6.19

D8/1 p.78

Valid values of [a ELLIPTICAL ARC CLOSE element's] vector components are those which produce vectors of non-zero length.

F74: Clause 5.7.1

D3/1 p.79

Legal values [of line bundle index] are positive integers.

F75: Clause 5.7.2

D5/1 p.79

Values [of line type] above 5 are reserved for registration and future standardization.

F76: Clause 5.7.3

D4/1 p.80

Valid values of 'line width specifier' are non-negative VDC if LINE WIDTH SPECIFICATION MODE is 'absolute' and non-negative reals if LINE WIDTH SPECIFICATION MODE is 'scaled.'

F77: Clause 5.7.5

D3/1 p.81

Legal values [of marker bundle index] are positive integers.

F78: Clause 5.7.5

D6/1 p.81

Values [of marker type] above 5 are reserved for registration and future standardization.

F79: Clause 5.7.7

D4/1 p.82

Valid values of 'marker size specifier' are non-negative VDC if MARKER SIZE SPECIFICATION MODE is 'absolute' and non-negative reals if MARKER SIZE SPECIFICATION MODE is 'scaled.'

F80: Clause 5.7.9

D3/1 p.83

Legal values [of text bundle index] are positive integers.

F81: Clause 5.7.10

D4/1 p.84

Legal values of the font index parameter are positive integers.

F82: Clause 5.7.12

D5/1 p.85

Legal values of the character expansion factor are non-negative reals.

F83: Clause 5.7.15
D3/1 p.87

Valid values of 'character height' are non-negative VDC.

F84: Clause 5.7.16
D2/1 p.87

Valid values of the [character up and character base] vectors include any which have non-zero length and are not collinear.

F85: Clause 5.7.19
D2/1 p.89

Legal values of [the] character set index parameter are positive integers.

F86: Clause 5.7.20
D2/1 p.89

Legal values of the alternate character set index parameter are positive integers.

F87: Clause 5.7.21
D2/1 p.90

Legal values of FILL BUNDLE INDEX are positive integers.

F88: Clause 5.7.22
D2/1 p.90

If other non-standardized values of interior style are used, they shall be given private values.

[NOTE: It is assumed that "private" in this context means "negative valued."]

F89: Clause 5.7.24
D7/1 p.91

Values [of hatch index] above 6 are reserved for registration and future standardization.

F90: Clause 5.7.25
D5/1 p.92

Legal values of PATTERN INDEX are positive integers.

F91: Clause 5.7.26
D2/1 p.92

Legal values of EDGE BUNDLE INDEX are positive integers.

F92: Clause 5.7.27
D5/1 p.93

Non-negative values of the [edge type] index [above 5] are reserved for [registration and] future standardization.

F93: Clause 5.7.28
D4/1 p.94

Valid values of 'edge width specifier' are non-negative VDC if EDGE WIDTH SPECIFICATION MODE is 'absolute' and non-negative reals if EDGE WIDTH SPECIFICATION MODE is 'scaled.'

F94: Clause 5.7.32
D2/1 p.96

Legal values of the pattern table index parameter are positive integers.

F95: Clause 5.6.32
D4/1-2 p.96

Legal values of the 'local colour precision' include the legal values of COLOUR (INDEX) PRECISION. In addition, each encoding defines a special value, the 'default colour precision indicator', as an indicator that the colour specifiers of the [PATTERN TABLE] element are to be encoded in the COLOUR (INDEX) PRECISION of the metafile; i.e., to indicate that the 'local colour precision' defaults to COLOUR (INDEX) PRECISION.

F96: Clause 5.7.33
D3/2 p.96

The pattern size vectors ... define a parallelogram.

[NOTE: This might be taken to imply that the vectors have non-zero length and are not collinear.]

F97: Clause 5.7.34
D2/1 p.97

Legal values of the colour index are non-negative integers.

F98: Clause 5.8.1
D1/3 p.99

Non-negative values [of the ESCAPE function identifier] are reserved for registration and future standardization.

[NOTE: This implies that 0 is not a valid identifier at this time.]

IV. REQUIREMENTS EXTRACTED FROM ISO 8632-3:1987

A. Explanation of Contents

The CGM Binary Encoding, ISO 8632-3:1987, was systematically read and all requirements relating to the conformance of instances of binary-encoded CGMs were extracted. Each separate statement of requirements is assigned a number and quoted. The requirement numbers are assigned sequentially from one and all start with the letter "B," indicating that these requirements come from the CGM Binary Encoding, Part 3.

A citation for each statement (shown in boldface) is given according to the following scheme: The Part 1 clause number is provided on the first line; on the second line, for Clause 4, 5, and 9 citations, the paragraph number (p) and sentence number(s) are specified as "#p/s", followed by the page number on which the text occurs. For Clause 6 citations, the only difference is that the second line starts with a Note number, rather than a "#". This means that the information is contained in a note on the specified page. Similarly, for Clause 7 citations, the only difference is that the second line starts with a Code number, rather than a "#". This means that the information is contained on the specified page in a note which expands upon the information in the tables in Clause 7.

Any commentary on the requirements statement is shown as a note enclosed in brackets ([NOTE:...]).

B. Organization of Requirements

The requirements are stated in order of their clause number.

C. Errors in the CGM Specification

Two errors were noted in ISO 8632-3:1987:

- (a) Clause 7.3 Table 4 (p.21): The parameter range for the first METAFILE ELEMENT LIST parameter should be +IR not ++IR.
- (b) Clause 7.3 Table 8 (p.32): The parameter range for CHARACTER EXPANSION FACTOR should be ++RR (not +RR) to be consistent with the other Parts of the CGM standard. However, it should be noted that ++RR is consistent with GKS (ISO 7942) and the other Parts of the CGM standard are not!

B1: Clause 4.3
#1/1 p.6

The binary encoding of the metafile is a logical data structure consisting of a sequential collection of bits.

B2: Clause 4.3
#3/2 p.6

Metafile elements are constrained to start on word boundaries within the binary data structure (this alignment may necessitate padding an element with bits to a word boundary if the parameter data of the element does not fill to such a boundary).

B3: Clause 4.4
#1/2 p.7

Metafile elements are represented in the Binary Encoding in one of two forms--short-form commands and long-form commands.

B4: Clause 4.4
#2/1 p.7

The short-form command always contains a complete element.

B5: Clause 4.4
#3/1 p.7

The short-form command only accommodates parameter lists up to 30 octets in length.

B6: Clause 4.4
#3/1 p.7

The long-form command accommodates lengths up to 32767 octets per data partition.

B7: Clause 4.4
#7/1-2 p.8

The first word of a long-form command is identical in structure to the first word of a short-form command. The presence of the value 11111 binary (decimal 31) in the parameter list length field indicates that the command is a long-form command.

B8: Clause 4.4
#7/3-5 p.8

The Command Header for the long-form command consists of two words. The second word contains the actual parameter list length. The two header words are then followed by the parameter list.

B9: Clause 4.4
#8/1-4 p.8

The long-form command allows the parameter list to be partitioned. Bit 15 of the second word indicates whether the given data complete the element or more data follow. For subsequent data partitions of the element, the first word of the long-form Command Header is omitted; only the second word, containing the parameter list length, is given.

B10: Clause 4.4
#8/5 p.8

The parameter list length for each partition specifies the length of that partition, not the length of the complete element.

B11: Clause 4.4
#8/56 p.8

The final partition of an element is indicated by bit 15 of the parameter list length word being zero.

B12: Clause 4.4
#10/7 p.8

Unless otherwise stated, the order of parameters is as listed in clause 5 of Part 1.

B13: Clause 4.4
#11/1-2 p.8

Every command is constrained to begin on a word boundary. This necessitates padding the command with a single null octet at the end of the command if the command contains an odd number of octets of parameter data.

B14: Clause 4.4
#11/3 p.9

In elements with parameters whose precisions are shorter than one octet (i.e., those containing a 'local colour precision' parameter) it is necessary to pad the last data-containing octet with null bits if the data do not fill the octet.

B15: Clause 4.4
#11/4 p.9

In all cases, the parameter list length is the count of octets actually containing parameter data--it does not include the padding octet if one is present. It is only at the end of a command that padding is performed, with the single exception of the CELL ARRAY element.

B16: Clause 4.4
#14/1 p.9

The short form command header with element class 15, element id 127, and parameter list length 0 is reserved for extension of the number of available element classes in future revisions [of the standard].

[NOTE: This particular element should not be encountered in version 1 metafiles.]

B17: Clause 5
#1/1-2 p.10

The Binary Encoding of the CGM uses five primitive data forms to represent the various abstract data types used to describe parameters in ISO 8632/1. [These are Signed Integer (SI), Unsigned Integer (UI), Character (C), Fixed Point Real (FX), and Floating Point Real (FP).]

B18: Clause 5
#5/2 p.10

In general, parameters may align on odd or even octet boundaries, because they may be preceded by an odd or even number of octets of other parameter data.

B19: Clause 5
#5/3-4 p.10

Elements containing the local colour precision parameter may have parameters shorter than one octet. It is possible in such cases that the parameters will not align on octet boundaries.

B20: Clause 5.1
#1/1 p.10

Signed integers are represented in "two's complement format."

B21: Clause 5.1

#1/2 p.10

Four precisions may be specified for signed integers: 8-bit, 16-bit, 24-bit, and 32-bit. Integer coordinate data encoded with this primitive data form do not use the 8-bit precision.

B22: Clause 5.2

#1/1 p.11

Four precisions may be specified for unsigned integers: 8-bit, 16-bit, 24-bit, and 32-bit.

B23: Clause 5.3

#1/1 p.12

Each character is stored in an octet [with the i-th character occupying the most significant octet of a word].

B24: Clause 5.4

#1/1 p.12

Fixed point real values are stored as two integers; the first represents the "whole part" and has the same form as a Signed Integer; the second represents the "fractional part" and has the same form as an Unsigned Integer.

B25: Clause 5.4

#1/2 p.12

Two precisions may be specified for Fixed Point Reals: 32-bit and 64-bit.

B26: Clause 5.4.3

#1/1 p.13

The values of the represented [fixed point] real numbers are given by [specific equations in this clause of the standard].

B27: Clause 5.5

#1/1 p.13

Floating Point Real values are represented in the floating point format of ANSI/IEEE 754.

B28: Clause 5.5

#2/2 p.13

Two precisions may be specified for Floating Point Reals: 32- bit or 64-bit.

B29: Clause 6

Note 3 p.16

Abstract parameter type Enumeration, E, is encoded identically to abstract type Index, IX, at 16-bit precision.

B30: Clause 6

Note 6 p.16-17

A string is encoded as a count (unsigned integer) followed by characters. The encoding of the count is similar to the encoding of length information for metafile commands themselves. If the first octet is in the range 0..254, then it represents the character count for the complete string. If the first octet is 255, then the next 16 bits contain the character count and a continuation flag. The first bit is used as a continuation flag (allowing strings longer than 32767 characters) and the next 15 bits represent the count, 0..32767, for the partial string. If the first bit is 0, then this partial string completes the string parameter. If 1, then this partial string will be followed by another.

[NOTE: This implies that a null string is encoded as a single octet with value 0.]

B31: Clause 6

Note 10 p.17

Fixed point reals apply to VDC and to R parameters for the following elements: line width, edge width, character spacing, character expansion factor, marker size, vertical continuous text alignment, horizontal continuous text alignment.

[NOTE: This implies that the metric scale factor parameter of the SCALING MODE element may be represented only as a Floating Point Real.]

B32: Clause 6

Note 11 p.17

For PACKED [CELL ARRAY] mode, each row of the cell array is represented by an array of colour values without compression. Each row starts on a word boundary.

B33: Clause 6
Note 11 p.17

For RUN LENGTH encoding, the data for each row [of the cell array] begins on a word boundary and consists of run-length-lists for runs of constant colour value.

B34: Clause 7.2
Code 0 p.20

A NO-OP has 1 parameter, P1, [which consists of] an arbitrary sequence of n octets, n=0,1,2,...

[NOTE: This implies that the minimum NO-OP element consists of two octets each with value 0.]

B35: Clause 7.3
Code 12 p.22

METAFILE DEFAULTS REPLACEMENT has 1 parameter that itself contains metafile elements. The structure and format is identical to appropriate metafile element(s).

[NOTE: This implies that all word alignment rules for metafile elements also apply to these elements when they are part of the parameter to METAFILE DEFAULTS REPLACEMENT.]

B36: Clause 7.4
Code 1 p.24

[The second] SCALING MODE parameter is a (real) metric scaling factor which is ignored if [the first parameter] is 0.

[NOTE: In Table 5, the real factor is always expressed as floating point (FP); fixed point is not allowed for this field or else the entry would have read R not FP.]

B37: Clause 7.6
Code 9 p.30

[In the eighth parameter of CELL ARRAY when cell representation mode is 'run length'] each list item consists of a cell count (integer) followed by a colour value.

B38: Clause 7.7

Code 32 p.36

[NOTE: NB: Unlike CELL ARRAY, pattern definitions in PATTERN TABLE have no "cell representation mode"; that is, they cannot be run-length encoded.]

B39: Clause 9

#2/1 p.41

[A metafile conforms to this encoding if] each metafile element is coded in the manner described.

B40: Clause 9

#3/1 p.41

[A metafile conforms to this encoding if] private metafile elements are all coded using the GDP or ESCAPE metafile elements as appropriate. Opcodes reserved for future standardization are not used to code private (non-standard) metafile elements.

[NOTE: This implies that a CGM is not conforming if it contains any opcodes (class/id pairs) not standardized in ISO 8632.]

B41: Clause 9

#4/1 p.41

[A metafile conforms to this encoding if] private values of index parameters are all coded using negative numbers.

B42: Clause 9

#5/1 p.41

[A metafile conforms to this encoding if] values specified as being "reserved for registration or future standardization" are not used unless their meaning has been registered.

[NOTE: Because no new elements have been registered or standardized, no such values should be encountered as yet.]

B43: Clause 9

#7/1 p.41

A conforming metafile may include, within the string parameters of TEXT, RESTRICTED TEXT, and APPEND TEXT elements, as well as string parameters within the data records of GDP elements, the ISO 2022 controls for designating and invoking G- sets. This is an alternative way, in addition to CHARACTER SET INDEX, by which character sets for displaying text strings may be selected.

B44: Annex C
all p.48-50

These are the codes used in the METAFILE ELEMENTS LIST element:

0/1 through 0/5
1/1 through 1/15
2/1 through 2/7
3/1 through 3/6
4/1 through 4/19
5/1 through 5/35
6/1
7/1 and 7/2.

[NOTE: Note the omission of 0/0 (NOOP). This requirement is stated in Annex C, but it merely summarizes the contents of the Clause 7 tables, so it should be upheld as a valid requirement.]

V. REQUIREMENTS EXTRACTED FROM MIL-D-28003

A. Explanation of Contents

The Military Specification, Digital Representation for Communication of Illustration Data: CGM Application Profile, MIL-D-28003 (20 December 1988), was systematically read and all requirements relating to the definition of a CALS conforming basic metafile were extracted. Each separate statement of requirements is assigned a number and quoted. The requirement numbers are assigned sequentially from one and all start with the letter "M," indicating that these requirements come from MIL-D-28003, the Milspec CGM Application Profile.

A citation for each statement (shown in boldface) is given according to the following scheme: The clause number is provided on the first line; on the second line, for most citations, the paragraph number (p) and sentence number(s) are specified as "#p/s", followed by the page number on which the text occurs. For a few citations, the only difference is that the second line starts with a Table number, rather than a "#". This means that the information is contained in the specified table on the specified page.

Any commentary on the requirements statement is shown as a note enclosed within brackets ([NOTE:...]).

B. Organization of Requirements

The requirements are grouped into eight main categories. Within each category, the requirements are generally stated in order of their clause number. The only exception to this rule is when a requirement is stated in several places in the standard. These requirement statements are grouped together.

The eight categories of requirements statements are:

- General CALS
- Physical to Logical Mapping
- Metafile Structure and Ordering
- Parameter Decoding
- Parameter Ranges
- Size Constraints
- GDP and ESCAPE Elements
- Colour/Pattern Usage Rules.

In addition, MIL-D-28003 corrects several editorial errors found in ANS X3.122 adopted by FIPS PUB 128. These corrections are placed in their appropriate category and marked with the phrase (CORRECTION) preceding the Clause number on the first line of the citation.

1. General CALS

M1: Clause 3.1

1/5 p.5

A conforming basic metafile shall contain no elements or parameters outside of the Basic Set.

M2: Clause 3.1.3

1/1 p.7

A conforming basic metafile shall not contain scalar values of parameter data outside the ranges specified by this specification.

M3: Clause 3.1.4

1/1 p.7

A conforming basic metafile shall use only the CGM Binary Encoding, as defined in FIPS PUB 128, part 3.

M4: Clause 3.2.1

1/1-2 p.7

The Basic Set shall be defined by the limitations on Basic Values [noted in subclauses of this clause]. Where an element is not mentioned, it is implied that the Basic Set shall include all values permitted in FIPS PUB 128.

M5: Clause 3.2.3

1/1 p.11

The defaults of all elements in this Application Profile shall be as specified in Clause 6 of Part 1 of FIPS PUB 128.

M6: Clause 3.2.3

1/2 p.11

Conforming basic metafiles are permitted to contain one or more METAFILE DEFAULTS REPLACEMENT elements to redefine any of these values.

2. Physical to Logical Mapping

M7: Clause 3.1.5

1/1 p.7

All basic metafiles conforming to this specification shall consist of 80-octet records.

M8: Clause 3.1.5

1/2 p.7

When [conforming basic meta]files are being transmitted on magnetic tape, the 80-octet logical records shall be blocked into 800-octet physical records.

3. Metafile Structure and Ordering

M9: (CORRECTION) Clause 3.1.6

5/1 p.7

Metafile Descriptor elements ... shall not be included in the METAFILE DEFAULTS REPLACEMENT [element].

M10: Clause 3.2.6.1

1/6-7 p.13

If used, this ESCAPE element (Disable clearing of view surface: id -301; data record null) must appear in the Metafile Descriptor. This ESCAPE element shall be a basic capability of the CGM Application Profile under this specification.

M11: Clause 3.2.6.2

1/6ff p.13

If used, this ESCAPE element (Device viewport: id -302; data record, a single string of text) must appear in the Picture Descriptor. This ESCAPE element shall be a basic capability of the CGM Application Profile under this specification.

M12: Clause 3.2.6.3

1/4ff p.14

This ESCAPE element (Implicit colour table: id -303; data record, a single string of text) shall be allowed in the Metafile Descriptor. This ESCAPE element shall be a basic capability of the CGM Application Profile under this specification.

M13: Clause 3.2.7.1

3/1-3 p.15

The METAFILE DEFAULTS REPLACEMENT element shall not be partitioned. Note that FIPS PUB 128 permits multiple occurrences of this element, so that partitioning is not required. Partitioning shall be permitted for all other elements.

4. Parameter Decoding

M14: (CORRECTION) Clause 3.1.6
3/1 p.7

Part 3, p.17, item 11: The fraction numerator which is "pn_x" should be "pn_x -1".

M15: (CORRECTION) Clause 3.1.6
4/1 p.7

Part 3, p.26, VDC REAL PRECISION: "3I" should be "E,2I".

M16: Clause 3.2.1.3
1/1-2 p.8

Note that the scale-factor parameter of SCALING MODE is always a floating-point number, even when REAL PRECISION has selected fixed point for other real numbers. It is not apparent in FIPS PUB 128 what the precision of this floating point parameter is when fixed point reals have been selected: its precision shall be (0,9,23).

5. Parameter Ranges

M17: Clause 3.2.1.1
1/1 p.8

The only constraint on delimiter elements shall be for NO-OP, and the basic values allowed shall be an arbitrary sequence of n octets, n=0..32767.

M18: Clause 3.2.1.2
Table I p.8

The METAFILE DESCRIPTION element's string (a) shall include a substring briefly identifying company or product [and] (b) shall contain the substring, "MIL-D-28003/BASIC-1".

M19: Clause 3.2.1.2
Table I p.8

INTEGER PRECISION Basic Value is 16.

M20: Clause 3.2.1.2
Table I p.8

REAL PRECISION Basic Values are (1,16,16) and (0,9,23).

M21: Clause 3.2.1.2
Table I p.8

INDEX PRECISION Basic Value is 16.

M22: Clause 3.2.1.2
Table I p.8

COLOUR PRECISION Basic Values are 8 and 16.

M23: Clause 3.2.1.2
Table I p.8

COLOUR INDEX PRECISION Basic Values are 8 and 16.

M24: Clause 3.2.1.2
Table I p.8

For FONT LIST, [up to] four simultaneous fonts are supported.
The font names are selected from those in [clause] 3.2.5 (q.v.).

M25: Clause 3.2.5
1/2-4 p.12

All of [the font names in Table VI] shall be considered basic capabilities of a basic metafile conforming to this specification. Any of these font [names] may appear in the FONT LIST element in a basic metafile that conforms to this specification. Font name shall be the concatenation of the string "HERSHEY:", to designate one of the Hershey fonts, and a "name string" to designate the particular typeface.

M26: Clause 3.2.5
Table VI p.12

The Basic font names are:

HERSHEY:CARTOGRAPHIC_ROMAN
HERSHEY:CARTOGRAPHIC_GREEK
HERSHEY:SIMPLEX_ROMAN
HERSHEY:SIMPLEX_GREEK
HERSHEY:SIMPLEX_SCRIPT
HERSHEY:COMPLEX_ROMAN
HERSHEY:COMPLEX_GREEK
HERSHEY:COMPLEX_SCRIPT
HERSHEY:COMPLEX_ITALIC
HERSHEY:COMPLEX_CYRILLIC
HERSHEY:DUPLEX_ROMAN
HERSHEY:TRIPLEX_ROMAN
HERSHEY:TRIPLEX_ITALIC
HERSHEY:GOTHIC_GERMAN
HERSHEY:GOTHIC_ENGLISH
HERSHEY:GOTHIC_ITALIAN

M27: Clause 3.2.1.2
Table I p.8

CHARACTER SET LIST Basic Value is a two-element list:
{(0,4/2),(1,4/1)}, which correspond to X3.4 (7-bit ASCII) and
X3.134/2 (8-bit ASCII).

M28: Clause 3.2.1.2
Table I p.8

CHARACTER CODING ANNOUNCER Basic Values are 0 (Basic 7-bit) and 1
(Basic 8-bit).

M29: Clause 3.2.1.4
Table II p.9

VDC INTEGER PRECISION Basic Values are 16 and 32.

M30: Clause 3.2.1.4
Table II p.9

VDC REAL PRECISION Basic Values are (1,16,16) (fixed) and
(0,9,23) (floating point).

M31: Clause 3.2.1.4
Table II p.9

TRANSPARENCY Basic Value is 1 (on).

M32: Clause 3.2.1.6
Table III p.9-10

LINE BUNDLE INDEX Basic Values are 1-5.

M33: Clause 3.2.1.6
Table III p.9-10

LINE TYPE Basic Values are 1-5 plus those defined in clause 3.2.2.1.

M34: Clause 3.2.2.1
Table IV p.10

[Ten] additional line types specified in table IV shall apply. Their CGM parameter values are the consecutive integers - 11301 through -11310.

M35: Clause 3.2.1.6
Table III p.9-10

MARKER BUNDLE INDEX Basic Values are 1-5.

M36: Clause 3.2.1.6
Table III p.9-10

MARKER TYPE Basic Values are 1-5.

M37: Clause 3.2.1.6
Table III p.9-10

TEXT BUNDLE INDEX Basic Values are 1-2.

M38: Clause 3.2.1.6
Table III p.9-10

TEXT FONT INDEX Basic Values are 1-4 and the character set selected shall be representable in the font selected.

M39: Clause 3.2.1.6
Table III p.9-10

CHARACTER SET INDEX Basic Values are 1-2 and the character set selected shall be representable in the font selected.

M40: Clause 3.2.1.6
Table III p.9-10

ALTERNATE CHARACTER SET INDEX Basic Values are 1-2 and the character set selected shall be representable in the font selected.

M41: Clause 3.2.1.6
Table III p.9-10

FILL BUNDLE INDEX Basic Values are 1-5.

M42: Clause 3.2.1.6
Table III p.9-10

HATCH INDEX Basic Values are 1-6 plus the hatch styles (indexes) defined in clause 3.2.2.2.

M43: Clause 3.2.2.2
Table V p.11

[Eighteen] additional hatch styles specified in table V shall apply. Their CGM parameter values are the consecutive integers -11401 through -11407 and -11409 through -11418.

M44: Clause 3.2.1.6
Table III p.9-10

EDGE BUNDLE INDEX Basic Values are 1-5.

M45: Clause 3.2.1.6
Table III p.9-10

EDGE TYPE Basic Values are 1-5.

M46: Clause 3.2.1.6
Table III p.9-10

PATTERN TABLE Basic Values are (Starting Index: 1-8; nx, ny: 1-16).

M47: Clause 3.2.1.6
Table III p.9-10

COLOUR TABLE Basic Values are those with start index in the range 0-255.

M48: Clause 3.2.1.8
1/1 p.10

The "action required" flag of the MESSAGE element shall be restricted to the value "no action required".

6. Size Constraints

M49: Clause 3.2.8.3

3/1 p.18

The basic value for the number of colour values that can appear in a colour array or colour list parameter shall be: 1048576 for CELL ARRAY (one 1024x1024 image); 2048 for PATTERN TABLE (eight 16x16 patterns); 256 for COLOUR TABLE (entries 0-255).

M50: Clause 3.2.8.3

5/1 p.18

The basic value for the number of points and VDC that can appear in parameters for metafile elements shall be 1024.

M51: Clause 3.2.8.3

7/1 p.18

The basic value for the length of an individual string of characters shall be: 254 for all string parameters except data records; 32767 for data records.

7. GDP and ESCAPE Elements

M52: Clause 3.2.1.5

1/1 p.9

Conforming basic metafiles shall not contain any Generalized Drawing Primitive (GDP) elements.

M53: Clause 3.2.1.7

1/1 p.10

CGM application profiles conforming to this specification may contain only those ESCAPE elements that are defined in 3.2.6.

M54: Clause 3.2.6.2
3/lff p.13-14

[For ESCAPE -302 the] escape data record [is] a single string of text containing the specification of the viewport. Parameters in the viewport shall be separated by at least one blank character and/or a single comma character. The decimal point of the real fraction shall be required. Leading zeros of the real fraction shall be optional. There are four parameters:

P1: First corner x-coordinate. Real fraction of the default device viewport, in the range [0.0,1.0].

P2: First corner y-coordinate. Real fraction of the default device viewport, in the range [0.0,1.0].

P4: Second corner x-coordinate. Real fraction of the default device viewport, in the range [0.0,1.0].

P4: Second corner y-coordinate. Real fraction of the default device viewport, in the range [0.0,1.0].

M55: Clause 3.2.6.3
1/5 p.14-15

The single integer parameter of the [-303] ESCAPE shall [be in the range] 0-2. The default value shall be 1.

M56: Clause 3.2.6.3
9/1-2 p.15

The [single] integer [parameter of the -303 ESCAPE shall be] encoded as "clear text," [e.g.,] value 2 is encoded as the string comprised of (or containing) the ASCII character "2".

8. Colour/Pattern Usage Rules

M57: Clause 3.2.1.6
5/1-3 p.10

For indexed colour selection, either all colour indexes used in the metafile shall have their representations defined by use of the COLOUR TABLE element, or none shall. A colour index is "used" if it occurs in an element selecting a colour value to be applied to a primitive (LINE COLOUR, CELL ARRAY, etc.). A colour index is also "used" if it is the default for a primitive attribute and the default applies to a displayed primitive.

M58: Clause 3.2.7.1
7/2 p.16

If a COLOUR TABLE element defining the representation of a given colour index appears in a picture, it shall appear before reference to that index by an attribute element or use of that index by a graphical primitive element (included in the latter shall be implicit use of default colour index attribute values by the first occurrence of an associated primitive).

M59: Clause 3.2.7.1
7/3 p.16

Once a given colour representation is defined and used, it shall not be redefined.

M60: Clause 3.2.7.1
9/2 p.16

If a PATTERN TABLE element defining the representation of a given pattern index appears in a picture: (a) it shall appear before explicit reference to that index by any PATTERN INDEX element; or (b) in the case of the default PATTERN INDEX, it shall appear before any implicit reference caused by the first occurrence of an associated filled primitive.

M61: Clause 3.2.7.1
9/3 p.16

Once a given pattern representation is defined and used, it shall not be redefined.

VI. CROSS REFERENCE TABLE BETWEEN REQUIREMENTS AND TESTING

<u>Requirement</u>	<u>Section Where Requirement is Represented for Testing</u>
F1	II, Step 6A
F2	II, Step 6A
F3	II, Step 6A
F4	II, Step 6A
F5	II, Step 4
F6	II, Step 4
F7	II, Step 5
F8	II, Step 5
F9	Appendix A
F10	Appendix A
F11	Appendix A
F12	Appendix A
F13	Appendix A
F14	Appendix A
F15	Appendix A
F16	Appendix A
F17	Appendix A
F18	Appendix A
F19	Appendix A
F20	Appendix A
F21	Appendix A
F22	II, Step 6B
F23	II, Step 6B
F24	Appendix C, COLOUR VALUE EXTENT

<u>Requirement</u>	<u>Section Where Requirement is Represented for Testing</u>
F25	Appendix C, COLOUR VALUE EXTENT
F26	Appendix C, colour index setting elements
F27	II, Step 5
F28	II, Step 5
F29	II, Step 5
F30	II, Step 5
F31	Appendix A
F32	Appendix A
F33	Appendix A
F34	Appendix A
F35	Appendix A
F36	Appendix A
F37	Appendix A
F38	Appendix A
F39	Appendix A
F40	Appendix A
F41	Appendix A
F42	Appendix A
F43	Appendix C, string parameters of text elements
F44	Appendix C, string parameters of text elements
F45	Appendix C, string parameters of text elements
F46	Appendix C, METAFILE ELEMENT LIST, & II, Step 5
F47	Appendix C, METAFILE ELEMENT LIST, & II, Step 5
F48	Appendix C, METAFILE ELEMENT LIST, & II, Step 5

<u>Requirement</u>	<u>Section Where Requirement is Represented for Testing</u>
F49	Appendix C, METAFILE ELEMENT LIST, and II, Steps 4B and 4C
F50	Appendix C, CELL ARRAY, and II, Steps 4B and 4C
F51	Appendix C, colour setting elements, CELL ARRAY, PATTERN TABLE, and II, Steps 4B and 4C
F52	Appendix C, VDC EXTENT
F53	Appendix C, CELL ARRAY, and II, Steps 4B and 4C
F54	Appendix C, PATTERN TABLE, & II, Steps 4B and 4C
F55	Appendix C, LINE TYPE, MARKER TYPE, HATCH STYLE, EDGE TYPE, FONT LIST, GDP, and ESCAPE
F56	Appendix C, index parameters
F57	Appendix C, METAFILE VERSION
F58	Appendix C, POLYLINE, and II, Step 5
F59	Appendix C, DISJOINT POLYLINE, and II, Step 5
F60	Appendix C, text elements
F61	Appendix C, CELL ARRAY
F62	Appendix C, CELL ARRAY
F63	Appendix C, GDP
F64	Appendix C, CIRCLE
F65	Appendix C, CIRCULAR ARC CENTRE
F66	Appendix C, CIRCULAR ARC CENTRE
F67	Appendix C, CIRCULAR ARC CENTRE CLOSE
F68	Appendix C, CIRCULAR ARC CENTRE CLOSE
F69	Appendix C, ELLIPSE
F70	Appendix C, ELLIPTICAL ARC

<u>Requirement</u>	<u>Section Where Requirement is Represented for Testing</u>
F71	Appendix C, ELLIPTICAL ARC
F72	Appendix C, ELLIPTICAL ARC CLOSE
F73	Appendix C, ELLIPTICAL ARC CLOSE
F74	Appendix C, LINE BUNDLE INDEX
F75	Appendix C, LINE TYPE
F76	Appendix C, LINE WIDTH
F77	Appendix C, MARKER BUNDLE INDEX
F78	Appendix C, MARKER TYPE
F79	Appendix C, MARKER SIZE
F80	Appendix C, TEXT BUNDLE INDEX
F81	Appendix C, FONT INDEX
F82	Appendix C, CHARACTER EXPANSION FACTOR
F83	Appendix C, CHARACTER HEIGHT
F84	Appendix C, CHARACTER ORIENTATION
F85	Appendix C, CHARACTER SET INDEX
F86	Appendix C, ALTERNATE CHARACTER SET INDEX
F87	Appendix C, FILL BUNDLE INDEX
F88	Appendix C, INTERIOR STYLE
F89	Appendix C, HATCH INDEX
F90	Appendix C, PATTERN INDEX
F91	Appendix C, EDGE BUNDLE INDEX
F92	Appendix C, EDGE TYPE
F93	Appendix C, EDGE WIDTH
F94	Appendix C, PATTERN TABLE INDEX

<u>Requirement</u>	<u>Section Where Requirement is Represented for Testing</u>
F95	Appendix C, PATTERN TABLE
F96	Appendix C, PATTERN SIZE
F97	Appendix C, COLOUR TABLE
F98	Appendix C, ESCAPE

<u>Requirement</u>	<u>Section Where Requirement is Represented for Testing</u>
B1	II, Step 1
B2	II, Step 2
B3	II, Step 2
B4	II, Step 2
B5	II, Step 2
B6	II, Step 2
B7	II, Step 2
B8	II, Step 2
B9	II, Step 2
B10	II, Step 2
B11	II, Step 2
B12	II, Step 4B
B13	II, Step 4B
B14	II, Step 4B
B15	II, Step 4B
B16	II, Step 3
B17	II, Step 4C
B18	II, Step 4B
B19	II, Step 4B
B20	II, Step 4C
B21	Appendix C, signed integers
B22	Appendix C, unsigned integers
B23	II, Step 4C
B24	Appendix C, fixed point reals
B25	Appendix C, fixed point reals

<u>Requirement</u>	<u>Section Where Requirement is Represented for Testing</u>
B26	Appendix C, fixed point reals
B27	Appendix C, floating point reals
B28	Appendix C, floating point reals
B29	Appendix C, enumeration type parameters
B30	Appendix C, strings
B31	Appendix C, fixed point reals
B32	II, Step 4C, for CELL ARRAY
B33	II, Step 4C, for CELL ARRAY
B34	Appendix C, NOOP
B35	Appendix C, METAFILE ELEMENT LIST, & II, Step 4
B36	Appendix C, SCALING MODE
B37	II, Step 4C, for CELL ARRAY
B38	II, Step 4C
B39	II, Step 3
B40	II, Step 3
B41	Appendix C, index parameters
B42	II, Step 4D
B43	Appendix C, string parameters
B44	Appendix C, METAFILE ELEMENT LIST

<u>Requirement</u>	<u>Section Where Requirement is Represented for Testing</u>
M1	II, Steps 3 and 4B
M2	II, Step 4D
M3	II, Steps 1 and 2
M4	II, Step 4D
M5	II, Step 5
M6	II, Step 5
M7	II, Step 1
M8	II, Step 1
M9	Appendix A, METAFILE DEFAULTS REPLACEMENT
M10	Appendix A, ESCAPE
M11	Appendix A, ESCAPE
M12	Appendix A, ESCAPE
M13	II, Steps 2 and 5
M14	Appendix B, CELL ARRAY
M15	Appendix B, VDC REAL PRECISION
M16	Appendix B, SCALING MODE, and II, Step 4D
M17	Appendix C, NOOP
M18	Appendix C, METAFILE DESCRIPTION
M19	Appendix C, INTEGER PRECISION
M20	Appendix C, REAL PRECISION
M21	Appendix C, INDEX PRECISION
M22	Appendix C, COLOUR PRECISION
M23	Appendix C, COLOUR INDEX PRECISION
M24	Appendix C, FONT LIST
M25	Appendix C, FONT LIST

<u>Requirement</u>	<u>Section Where Requirement is Represented for Testing</u>
M26	Appendix C, FONT LIST
M27	Appendix C, CHARACTER SET LIST
M28	Appendix C, CHARACTER CODING ANNOUNCER
M29	Appendix C, VDC INTEGER PRECISION
M30	Appendix C, VDC REAL PRECISION
M31	Appendix C, TRANSPARENCY
M32	Appendix C, LINE BUNDLE INDEX
M33	Appendix C, LINE TYPE
M34	Appendix C, LINE TYPE
M35	Appendix C, MARKER BUNDLE INDEX
M36	Appendix C, MARKER TYPE
M37	Appendix C, TEXT BUNDLE INDEX
M38	Appendix C, TEXT FONT INDEX
M39	Appendix C, CHARACTER SET INDEX
M40	Appendix C, ALTERNATE CHARACTER SET INDEX
M41	Appendix C, FILL BUNDLE INDEX
M42	Appendix C, HATCH INDEX
M43	Appendix C, HATCH INDEX
M44	Appendix C, EDGE BUNDLE INDEX
M45	Appendix C, EDGE TYPE
M46	Appendix C, PATTERN TABLE
M47	Appendix C, COLOUR TABLE
M48	Appendix C, MESSAGE
M49	Appendix C, CELL ARRAY, PATTERN TABLE, and COLOUR TABLE

<u>Requirement</u>	<u>Section Where Requirement is Represented for Testing</u>
M50	Appendix C, point lists
M51	Appendix C, string parameters
M52	Appendix C, GDP, and II, Step 5
M53	Appendix C, GDP, and II, Step 5
M54	Appendix C, ESCAPE
M55	Appendix C, ESCAPE
M56	Appendix C, ESCAPE
M57	II, Steps 4B and 4C
M58	II, Step 5
M59	II, Step 5
M60	II, Step 5
M61	II, Step 5

APPENDIX A
CGM ELEMENT ORDERING REQUIREMENTS

Explanation of Contents

The CGM Functional Description, ISO 8632-1:1987, contains requirements regarding the permitted order of elements in a CGM. These requirements can best be summarized by describing a finite-state machine.

In the following, the states of the finite-state machine representing a CGM conformance-checking parser are listed, elements which cause state transitions are indicated, and which elements are valid in which states are indicated.

Seven states are needed:

MFCL	Metafile closed.
MDOP	Metafile Descriptor open, but first picture not yet open, and not processing a METAFILE DEFAULTS REPLACEMENT element.
MMDR	Metafile Descriptor open, but first picture not yet open; however, are processing a METAFILE DEFAULTS REPLACEMENT element.
PDOP	Picture Descriptor open, but not in Picture Body yet.
PBOP	Picture Body open, but not processing a "not final" TEXT or "not final" RESTRICTED TEXT element.
TXOP	Processing a "not final" TEXT or "not final" RESTRICTED TEXT element.
PICL	First or subsequent picture closed, but metafile not yet closed.

In the following, all CGM elements are listed along with the states in which it is valid for them to occur. Any elements causing state transitions are annotated as (***), and are in boldface. The binary encoding element class and id codes are also provided.

MIL-D-28003, the CALS CGM Application Profile, places a few additional constraints on GDPs and ESCAPE elements. These constraints are noted with the phrase (CALS) in the left margin and highlighted in boldface.

0, 0	NO-OPERATION	MDOP MMDR PDOP
0, 1	BEGIN METAFILE	PBOP TXOP PICL
		MFCL
(***)	Causes transition from MFCL to MDOP.	
0, 2	END METAFILE	MDOP
		PICL
(***)	Causes transition from MDOP to MFCL.	
(***)	Causes transition from PICL to MFCL.	
0, 3	first BEGIN PICTURE	MDOP
(***)	Causes transition from MDOP to PDOP.	
0, 3	second or later BEGIN PICTURE	
		PICL
(***)	Causes transition from PICL to PDOP.	
0, 4	BEGIN PICTURE BODY	PDOP
(***)	Causes transition from PDOP to PBOP.	
0, 5	END PICTURE	
		PBOP
(***)	Causes transition from PBOP to PICL.	
1, 1	METAFILE VERSION	MDOP
1, 2	METAFILE DESCRIPTION	MDOP
1, 3	VDC TYPE	MDOP
1, 4	INTEGER PRECISION	MDOP
1, 5	REAL PRECISION	MDOP
1, 6	INDEX PRECISION	MDOP
1, 7	COLOUR PRECISION	MDOP
1, 8	COLOUR INDEX PRECISION	MDOP
1, 9	MAXIMUM COLOUR INDEX	MDOP

1, 10	COLOUR VALUE EXTENT	MDOP
1, 11	METAFILE ELEMENT LIST	MDOP
1, 12	METAFILE DEFAULTS REPLACEMENT	MDOP
(***) Causes transition from MDOP to MMDR.		
(***) completion of last element contained in the METAFILE DEFAULTS REPLACEMENT parameter list causes transition from MMDR to MDOP.		
1, 13	FONT LIST	MDOP
1, 14	CHARACTER SET LIST	MDOP
1, 15	CHARACTER CODING ANNOUNCER	MDOP
2, 1	SCALING MODE	MMDR PDOP
2, 2	COLOUR SELECTION MODE	MMDR PDOP
2, 3	LINE WIDTH SPECIFICATION MODE	MMDR PDOP
2, 4	MARKER SIZE SPECIFICATION MODE	MMDR PDOP
2, 5	EDGE WIDTH SPECIFICATION MODE	MMDR PDOP
2, 6	VDC EXTENT	MMDR PDOP
2, 7	BACKGROUND COLOUR	MMDR PDOP
3, 1	VDC INTEGER PRECISION	MMDR PBOP
3, 2	VDC REAL PRECISION	MMDR PBOP
3, 3	AUXILIARY COLOUR	MMDR PBOP TXOP
3, 4	TRANSPARENCY	MMDR PBOP TXOP
3, 5	CLIP RECTANGLE	MMDR PBOP

3, 6	CLIP INDICATOR	MMDR
		PBOP
4, 1	POLYLINE	
		PBOP
4, 2	DISJOINT POLYLINE	
		PBOP
4, 3	POLYMARKER	
		PBOP
4, 4	"final" TEXT	
		PBOP
4, 4	"not final" TEXT	
		PBOP TXOP
(***)	Causes transition from PBOP to TXOP.	
4, 5	"final" RESTRICTED TEXT	
		PBOP
4, 5	"not final" RESTRICTED TEXT	
		PBOP TXOP
(***)	Causes transition from PBOP to TXOP.	
4, 6	"not final" APPEND TEXT	
		TXOP
4, 6	"final" APPEND TEXT	
		TXOP
(***)	Causes transition from TXOP to PBOP.	
4, 7	POLYGON	
		PBOP
4, 8	POLYGON SET	
		PBOP
4, 9	CELL ARRAY	
		PBOP

4,10	GENERALIZED DRAWING PRIMITIVE	PBOP
(CALS) This element is not permitted in a basic conforming metafile.		
4,11	RECTANGLE	PBOP
4,12	CIRCLE	PBOP
4,13	CIRCULAR ARC 3 POINT	PBOP
4,14	CIRCULAR ARC 3 POINT CLOSE	PBOP
4,15	CIRCULAR ARC CENTRE	PBOP
4,16	CIRCULAR ARC CENTER CLOSE	PBOP
4,17	ELLIPSE	PBOP
4,18	ELLIPTICAL ARC	PBOP
4,19	ELLIPTICAL ARC CLOSE	PBOP
5, 1	LINE BUNDLE INDEX	MMDR PBOP
5, 2	LINE TYPE	MMDR PBOP
5, 3	LINE WIDTH	MMDR PBOP
5, 4	LINE COLOUR	MMDR PBOP
5, 5	MARKER BUNDLE INDEX	MMDR PBOP

5, 6	MARKER TYPE	MMDR PBOP
5, 7	MARKER SIZE	MMDR PBOP
5, 8	MARKER COLOUR	MMDR PBOP
5, 9	TEXT BUNDLE INDEX	MMDR PBOP TXOP
5,10	TEXT FONT INDEX	MMDR PBOP TXOP
5,11	TEXT PRECISION	MMDR PBOP TXOP
5,12	CHARACTER EXPANSION FACTOR	MMDR PBOP TXOP
5,13	CHARACTER SPACING	MMDR PBOP TXOP
5,14	TEXT COLOUR	MMDR PBOP TXOP
5,15	CHARACTER HEIGHT	MMDR PBOP TXOP
5,16	CHARACTER ORIENTATION	MMDR PBOP
5,17	TEXT PATH	MMDR PBOP
5,18	TEXT ALIGNMENT	MMDR PBOP
5,19	CHARACTER SET INDEX	MMDR PBOP TXOP
5,20	ALTERNATE CHARACTER SET INDEX	MMDR PBOP TXOP
5,21	FILL BUNDLE INDEX	MMDR PBOP

5,22	INTERIOR STYLE	MMDR PBOP
5,23	FILL COLOUR	MMDR PBOP
5,24	HATCH INDEX	MMDR PBOP
5,25	PATTERN INDEX	MMDR PBOP
5,26	EDGE BUNDLE INDEX	MMDR PBOP
5,27	EDGE TYPE	MMDR PBOP
5,28	EDGE WIDTH	MMDR PBOP
5,29	EDGE COLOUR	MMDR PBOP
5,30	EDGE VISIBILITY	MMDR PBOP
5,31	FILL REFERENCE POINT	MMDR PBOP
5,32	PATTERN TABLE	MMDR PBOP
5,33	PATTERN SIZE	MMDR PBOP
5,34	COLOUR TABLE	MMDR PBOP
5,35	ASPECT SOURCE FLAGS	MMDR PBOP

6, 1	ESCAPE	MDOP MMDR PDOP
		PBOP TXOP PICL

(CALS) Only the following three ESCAPEs are permitted in a basic conforming metafile and only in the specified states.

-301 Disable viewsurface clear	MDOP	
-302 Device viewport		PDOP
-303 Implicit colour table	MDOP	

7, 1	MESSAGE	MDOP MMDR PDOP
		PBOP PICL

7, 2	APPLICATION DATA	MDOP MMDR PDOP
		PBOP PICL

APPENDIX B
CGM ELEMENT LENGTH REQUIREMENTS
FROM
ISO 8632-3:1987

The CGM Binary Encoding, ISO 8632-3:1987, Clause 7, contains tables showing, for each CGM element, the lengths of their parameter lists. Length is expressed as the number of octets occupied by the entire parameter list, but excluding the command header and any padding needed to provide word alignment.

The length of each parameter list is expressed as a function of a number of variables--among them are the various types and precisions allowed for each parameter. For an arbitrary combination of types and precisions, the table is rather complex to read and interpret. However, MIL-D-28003, the CGM Application Profile for CALS, greatly restricts the number of types and precisions that are permitted to occur in a basic conforming metafile. Consequently, it is feasible to develop a parameter length table with many elements having a fixed length and others limited to only a few variations.

Variability in length is due to five major causes:

- (1) VDC Type: If integer (I), all VDCs are 2 octets in length; if real (R), 4 octets in length.
- (2) Colour Selection Mode and Precision: If indexed, colours are 1 or 2 octets in length; if direct, they are 3 or 6 octets, depending upon the value of colour index (CX8/CX16) and colour precision (CD8/CD16).
- (3) String Length: Strings are either $n+1$ or $n+3$ octets long (depending on whether they are coded using the short form or the long form); because all strings in CALS (except the APPLICATION DATA data record) must contain fewer than 255 octets, for the purposes of this table it is assumed that all strings are coded with the short form.
- (4) Number of Items in List: Several of the geometric primitive elements and some of the other elements (e.g., FONT LIST and ASPECT SOURCE FLAGS) consist of a list of elements, whose count is not directly available. The length is clearly a function of the number of items in the list.
- (5) Local Colour Precision: The numerous values allowed for local colour precision in the CELL ARRAY and PATTERN TABLE elements make for great variation in the length of the parameter list for these elements.

In the following, all CGM elements are listed along with their lengths in octets. Where the length is a function of other variables, an expression giving the length in terms of these other variables is shown and a code is appended to show the source of variability. In the table that follows, n is consistently used to represent the number of characters (octets) in a string and m to represent the number of items in a list; e.g., for the number of points in a "poly" geometric primitive. The binary encoding element class and id codes are also provided in the table.

0, 0	NO-OPERATION	m
0, 1	BEGIN METAFILE	n + 1
0, 2	END METAFILE	0
0, 3	BEGIN PICTURE	n + 1
0, 4	BEGIN PICTURE BODY	0
0, 5	END PICTURE	0
1, 1	METAFILE VERSION	2
1, 2	METAFILE DESCRIPTION	n + 1
1, 3	VDC TYPE	2
1, 4	INTEGER PRECISION	2
1, 5	REAL PRECISION	6
1, 6	INDEX PRECISION	2
1, 7	COLOUR PRECISION	2
1, 8	COLOUR INDEX PRECISION	2
1, 9	MAXIMUM COLOUR INDEX	1 (CX8) 2 (CX16)
1,10	COLOUR VALUE EXTENT	6 (CD8) 12 (CD16)
1,11	METAFILE ELEMENT LIST	4m + 2
1,12	METAFILE DEFAULTS REPLACEMENT	variable
1,13	FONT LIST	m * (n+1)
1,14	CHARACTER SET LIST	m * (n + 3)
1,15	CHARACTER CODING ANNOUNCER	2

2, 1	SCALING MODE	6
2, 2	COLOUR SELECTION MODE	2
2, 3	LINE WIDTH SPECIFICATION MODE	2
2, 4	MARKER SIZE SPECIFICATION MODE	2
2, 5	EDGE WIDTH SPECIFICATION MODE	2
2, 6	VDC EXTENT	8 (I) 16 (R)
2, 7	BACKGROUND COLOUR	3 (CD8) 6 (CD16)
3, 1	VDC INTEGER PRECISION	2
3, 2	VDC REAL PRECISION	6
3, 3	AUXILIARY COLOUR	3 (CD8) 6 (CD16)
3, 4	TRANSPARENCY	2
3, 5	CLIP RECTANGLE	8 (I) 16 (R)
3, 6	CLIP INDICATOR	2
4, 1	POLYLINE	4m (I) 8m (R)
4, 2	DISJOINT POLYLINE	4m (I) 8m (R)
4, 3	POLYMARKER	4m (I) 8m (R)
4, 4	TEXT	7 + n (I) 11 + n (R)
4, 5	RESTRICTED TEXT	11 + n (I) 19 + n (R)
4, 6	APPEND TEXT	3 + n

4, 7	POLYGON	4m (I) 8m (R)
4, 8	POLYGON SET	6m (I) 10m (R)
4, 9	CELL ARRAY	20 + colour values (I) 32 + colour values (R)
4,10	GENERALIZED DRAWING PRIMITIVE	5 + 4m + n (I) 5 + 8m + n (R)
4,11	RECTANGLE	8 (I) 16 (R)
4,12	CIRCLE	6 (I) 10 (R)
4,13	CIRCULAR ARC 3 POINT	12 (I) 24 (R)
4,14	CIRCULAR ARC 3 POINT CLOSE	14 (I) 26 (R)
4,15	CIRCULAR ARC CENTRE	14 (I) 28 (R)
4,16	CIRCULAR ARC CENTER CLOSE	16 (I) 30 (R)
4,17	ELLIPSE	12 (I) 24 (R)
4,18	ELLIPTICAL ARC	20 (I) 32 (R)
4,19	ELLIPTICAL ARC CLOSE	22 (I) 34 (R)
5, 1	LINE BUNDLE INDEX	2
5, 2	LINE TYPE	2
5, 3	LINE WIDTH	2 (I) 4 (R)

5, 4	LINE COLOUR	1 (CX8) 2 (CX16) 3 (CD8) 6 (CD16)
5, 5	MARKER BUNDLE INDEX	2
5, 6	MARKER TYPE	2
5, 7	MARKER SIZE	2 (I) 4 (R)
5, 8	MARKER COLOUR	1 (CX8) 2 (CX16) 3 (CD8) 6 (CD16)
5, 9	TEXT BUNDLE INDEX	2
5,10	TEXT FONT INDEX	2
5,11	TEXT PRECISION	2
5,12	CHARACTER EXPANSION FACTOR	4
5,13	CHARACTER SPACING	4
5,14	TEXT COLOUR	1 (CX8) 2 (CX16) 3 (CD8) 6 (CD16)
5,15	CHARACTER HEIGHT	2 (I) 4 (R)
5,16	CHARACTER ORIENTATION	8 (I) 16 (R)
5,17	TEXT PATH	2
5,18	TEXT ALIGNMENT	12
5,19	CHARACTER SET INDEX	2
5,20	ALTERNATE CHARACTER SET INDEX	2
5,21	FILL BUNDLE INDEX	2

5,22	INTERIOR STYLE	2
5,23	FILL COLOUR	1 (CX8) 2 (CX16) 3 (CD8) 6 (CD16)
5,24	HATCH INDEX	2
5,25	PATTERN INDEX	2
5,26	EDGE BUNDLE INDEX	2
5,27	EDGE TYPE	2
5,28	EDGE WIDTH	2 (I) 4 (R)
5,29	EDGE COLOUR	1 (CX8) 2 (CX16) 3 (CD8) 6 (CD16)
5,30	EDGE VISIBILITY	2
5,31	FILL REFERENCE POINT	4 (I) 8 (R)
5,32	PATTERN TABLE	8 + colour values
5,33	PATTERN SIZE	8 (I) 16 (R)
5,34	COLOUR TABLE	1 + 3m (CX8,CD8) 2 + 3m (CX16,CD8) 1 + 6m (CX8,CD16) 2 + 6m (CX16,CD16)
5,35	ASPECT SOURCE FLAGS	4m
6, 1	ESCAPE	3 + n
7, 1	MESSAGE	3 + n
7, 2	APPLICATION DATA	3 + n (short DR) 5 + n (long DR)

APPENDIX C
ALLOWABLE PARAMETER RANGES
FOR
BINARY ENCODED CGM ELEMENTS

<u>Element</u>	<u>Parameter</u>	<u>Parameter Range</u>
NOOP	number of null-valued octets	>= 0
(CALS)		<= 32767
BEGIN METAFILE	length of string (P1) contents of string (P1)	>= 0 {set of legal character codes}
(CALS)	length of string	<= 254
END METAFILE	n/a	
BEGIN PICTURE	length of string (P1) contents of string (P1)	>= 0 {set of legal character codes}
(CALS)	length of string	<= 254
BEGIN PICTURE BODY	n/a	
END PICTURE n/a		
METAFILE VERSION	version number (P1)	= 1
METAFILE DESCRIPTION	length of string (P1) contents of string (P1)	>= 0 {set of legal character codes}
(CALS)	length of string contents of string	<= 254 contains "MIL-D-28003/BASIC-1" and a product name
VDC TYPE	type (P1)	{0,1}
INTEGER PRECISION	precision (P1)	{8,16,24,32}
(CALS)		{16}
REAL PRECISION	format (P1) field width 1 (P2) field width 2 (P3)	{0 9 23,0 12 52, 1 16 16,1 32 32}
(CALS)		{0 9 23,1 16 16}

<u>Element</u>	<u>Parameter</u>	<u>Parameter Range</u>
INDEX PRECISION	precision (P1)	{8,16,24,32}
(CALS)		{16}
COLOUR PRECISION	precision (P1)	{8,16,24,32}
(CALS)		{8, 16}
COLOUR INDEX PRECISION	precision (P1)	{8,16,24,32}
(CALS)		{8, 16}
MAXIMUM COLOUR INDEX	index value (P1)	> 0
(CALS)		<256
COLOUR VALUE EXTENT	minRed minGreen minBlue maxRed maxGreen maxBlue	>= 0 > corresponding minimum values
METAFILE ELEMENT LIST	number of elements (P1) for each (class/id) in the list of elements	> 0 One of the valid class/id pairs or -1/0 or -1,1
METAFILE DEFAULTS REPLACEMENT	n/a; however, within the span of the MDR element only completely specified Picture Descriptor, Control, Attribute, and Escape elements are allowed.	
FONT LIST	for each string in the list: length of string (P1) contents of string	>= 0 {set of legal character codes}
(CALS)	number of font names in list length of each string	<= 4 <= 254
CHARACTER SET LIST	for each item in list: type length of string (P2)	>= 0 AND <= 4 or negative > 0
(CALS)		{(0,4/2),(1,4/1)}

<u>Element</u>	<u>Parameter</u>	<u>Parameter Range</u>
CHARACTER CODING ANNOUNCER	coding technique	<= 3
(CALS)		{0,1}
SCALING MODE	mode (P1) metric scale factor (P2)	{0,1} {> 0.0 if P1=1}
(CALS)	metric scale factor (P2)	must be represented as Floating Point if mode is 1
COLOUR SELECTION MODE	mode (P1)	{0,1}
LINE WIDTH SPECIFICATION MODE	mode (P1)	{0,1}
MARKER SIZE SPECIFICATION MODE	mode (P1)	{0,1}
EDGE WIDTH SPECIFICATION MODE	mode (P1)	{0,1}
VDC EXTENT	n/a	
BACKGROUND COLOUR	Red Green Blue	each component >= corresponding min. colour extent AND <= corresponding max. colour extent
VDC INTEGER PRECISION	precision (P1)	{16,24,32}
(CALS)		{16, 32}
VDC REAL PRECISION	format (P1) field width 1 (P2) field width 2 (P3)	{0 9 23,0 12 52, 1 16 16,1 32 32}
(CALS)		{0 9 23,1 16 16}

<u>Element</u>	<u>Parameter</u>	<u>Parameter Range</u>
AUXILIARY COLOUR	colour index (P1) or Red Green Blue	<= maximum colour index each component >= corresponding min. colour extent AND <= corresponding max. colour extent
TRANSPARENCY	indicator (P1)	{0,1}
(CALS)		{1}
CLIP RECTANGLE	n/a	
CLIP INDICATOR	indicator (P1)	{0,1}
POLYLINE	n/a	[would like number of points >= 2]
(CALS)	number of points	<= 1024
DISJOINT POLYLINE	n/a	[would like number of points to be even AND >= 2]
(CALS)	number of points	<= 1024
POLYMARKER	n/a	
(CALS)	number of points	<= 1024
TEXT	flag length of string (P1) contents of string	{0,1} >= 0 {set of legal character codes}
(CALS)	length of string	<= 254
RESTRICTED TEXT	flag length of string (P1) contents of string	{0,1} >= 0 {set of legal character codes}
(CALS)	length of string	<= 254

<u>Element</u>	<u>Parameter</u>	<u>Parameter Range</u>
APPEND TEXT flag	length of string (P1) contents of string	{0,1} >= 0 {set of legal character codes}
(CALS)	length of string	<= 254
POLYGON	n/a	[would like number of points >= 3]
(CALS)	number of points	<= 1024
POLYGON SET	for each pair of values in list: edge out flag for each polygon in set	{0,1,2,3} [would like number of points >= 3]
(CALS)	number of points	<= 1024
CELL ARRAY	P, Q, R (P1, P2, P3) nx (P4) ny (P5) local colour precision (P6) cell rep. mode (P7) for each cell colour value: colour index or Red Green Blue	RQ x RP /= 0 >= 1 >= 1 { 0, 1, 2, 4, 8,16,24,32} {0,1} > 0 AND <= max. colour index each component >= corresponding min. colour extent AND <= corresponding max. colour extent
(CALS)	nx (P4) ny (P5)	<= 1024 <= 1024
GDP	identifier (P1) number of points (P2) length of data record string	<= m, where m = highest registered id. [m=-1 now] >= 0 >= 0
(CALS)	No GDP is allowed.	
RECTANGLE	n/a	

<u>Element</u>	<u>Parameter</u>	<u>Parameter Range</u>
CIRCLE	radius (P2)	≥ 0.0
CIRCULAR ARC	n/a	
CIRCULAR ARC 3 POINT	n/a	
CIRCULAR ARC 3 POINT CLOSE	close type (P4)	{0,1}
CIRCULAR ARC CENTRE	Start vector (P2) End vector (P3) radius (P4)	$\begin{vmatrix} P2 \\ P3 \end{vmatrix} \neq 0$ ≥ 0.0
CIRCULAR ARC CENTRE CLOSE	Start vector (P2) End vector (P3) radius (P4) close type (P5)	$\begin{vmatrix} P2 \\ P3 \end{vmatrix} \neq 0$ ≥ 0.0 {0,1}
ELLIPSE	Center (P1) First CDP (P2) Second CDP (P3)	$P1 \neq P2$ $P2 \neq P3$ $P3 \neq P1$
ELLIPTICAL ARC	Center (P1) First CDP (P2) Second CDP (P3) Start vector (P4) End vector (P5)	$P1 \neq P2$ $P2 \neq P3$ $P3 \neq P1$ $\begin{vmatrix} P4 \\ P5 \end{vmatrix} \neq 0$
ELLIPTICAL ARC CLOSE	Center (P1) First CDP (P2) Second CDP (P3) Start vector (P4) End vector (P5) close type (P6)	$P1 \neq P2$ $P2 \neq P3$ $P3 \neq P1$ $\begin{vmatrix} P4 \\ P5 \end{vmatrix} \neq 0$ {0,1}
LINE BUNDLE INDEX	index (P1)	> 0
(CALS)		{1..5}
LINE TYPE	type (P1)	< 0 OR {1..m} where m = highest registered or standardized line type. [m=5 now].
(CALS)		{1..5 and -11301...-11310}

<u>Element</u>	<u>Parameter</u>	<u>Parameter Range</u>
LINE WIDTH	VDC width or scale factor	≥ 0 or ≥ 0.0 depending on VDC type and specification mode
LINE COLOUR	colour index or Red Green Blue	\leq maximum colour index each component \geq corresponding min. colour extent AND \leq corresponding max. colour extent
MARKER BUNDLE INDEX	index (P1)	> 0
(CALS)		{1..5}
MARKER TYPE	type (P1)	< 0 OR {1..m} where m = highest registered or standard marker type. [m=5 now].
(CALS)		{1..5}
MARKER SIZE	VDC size or scale factor	≥ 0 or ≥ 0.0 depending on VDC type and specification mode
MARKER COLOUR	colour index or Red Green Blue	\leq maximum colour index each component \geq corresponding min. colour extent AND \leq corresponding max. colour extent
TEXT BUNDLE INDEX	index (P1)	> 0
(CALS)		{1,2}
TEXT FONT INDEX	index (P1)	> 0 [perhaps AND \leq number of fonts in font list]

<u>Element</u>	<u>Parameter</u>	<u>Parameter Range</u>
(CALS)		{1..4}
TEXT PRECISION	precision (P1)	{0,1,2}
CHARACTER EXPANSION FACTOR	expansion factor (P1)	>= 0.0
CHARACTER SPACING	n/a	
TEXT COLOUR	colour index or Red Green Blue	<= maximum colour index each component >= corresponding min. colour extent AND <= corresponding max. colour extent
CHARACTER HEIGHT	VDC height (P1)	>= 0 or >= 0.0 depending on VDC type
CHARACTER ORIENTATION	character up vector (P1) character base vector (P2)	P1 /= 0 P2 /= 0 up x base /= 0
TEXT PATH	path (P1)	{0,1,2,3}
TEXT ALIGNMENT	horizontal alignment (P1) vertical alignment (P2)	{0,1,2,3,4} {0,1,2,3,4,5,6}
CHARACTER INDEX	index (P1)	> 0 [perhaps AND SET <= number in character set list]
(CALS)		{1,2}
ALTERNATE CHARACTER SET INDEX	index (P1)	> 0 [perhaps AND <= number in character set list]
(CALS)		{1,2}
FILL BUNDLE INDEX	index (P1)	> 0

<u>Element</u>	<u>Parameter</u>	<u>Parameter Range</u>
(CALS)		{1..5}
INTERIOR STYLE	style (P1)	<= 4
FILL COLOUR	colour index or Red Green Blue	<= maximum colour index each component >= corresponding min. colour extent AND <= corresponding max. colour extent
HATCH INDEX	index (P1)	< 0 OR {1..m} where m = highest registered or standard hatch index. [m=6 now].
(CALS)		{1..6 and -11401..-11407, -11409..-11418}
PATTERN INDEX	index (P1)	> 0
EDGE BUNDLE INDEX	index (P1)	> 0
(CALS)		{1..5}
EDGE TYPE	type (P1)	< 0 OR {1..m} where m = highest registered or standardized edge type. [m=5 now].
(CALS)		{1..5}
EDGE WIDTH	VDC width or scale factor	>= 0 or >= 0.0 depending on VDC type and specification mode

<u>Element</u>	<u>Parameter</u>	<u>Parameter Range</u>
EDGE COLOUR	colour index or Red Green Blue	<= maximum colour index each component >= corresponding min. colour extent AND <= corresponding max. colour extent
EDGE VISIBILITY	visibility (P1)	{0,1}
FILL REFERENCE POINT	n/a	
PATTERN TABLE	index (P1) nx (P2) ny (P3) local colour precision (P4) for each cell colour value: colour index or Red Green Blue	> 0 >= 1 >= 1 { 0, 1, 2, 4, 8,16,24,32} <= maximum colour index each component >= corresponding min. colour extent AND <= corresponding max. colour extent
(CALS)	index (P1) nx (P2) ny (P3)	1 <= and <= 8 1 <= and <= 16 1 <= and <= 16
PATTERN SIZE	pattern height vector (P1) pattern width vector (P2)	P1 /= 0 P2 /= 0 height x width /=0
COLOUR TABLE	starting colour index for each colour to be loaded: Red Green Blue	<= maximum colour index each component >= corresponding min. colour extent AND <= corresponding max. colour extent

<u>Element</u>	<u>Parameter</u>	<u>Parameter Range</u>
	also	total-number-of colours-loaded plus starting- colour-index - 1 should not > max. colour index
(CALS)	starting colour index	< 256
ASPECT SOURCE FLAGS	for each pair of values: type value	{ 0 .. 17 } { 0,1 }
ESCAPE	identifier (P1)	<= m, where m = highest registered id. [m=-1 now]
	length of data record string	>= 0
(CALS)	identifier content of data record is prescribed	{ -301..-303 }
MESSAGE	action flag (P1)	{ 0,1 }
	len. of message string (P2)	>= 0
(CALS)	action flag	{ 1 }
APPLICATION DATA	length of data record string	>= 0
(CALS)	length of data record string	<= 32767

KEY:

|| Px || means the mathematical NORM of the vector specified
in parameter Px.

vector1 x vector2 means the vector cross product

[...] text in the square brackets contains additional
information for the tester.

/= means not equal to

<= means less than or equal to

>= means greater than or equal to

FINAL REPORT

CALS FY89 SOW TASK 4.1.3

INJECTION OF CALS REQUIREMENTS

IN EXTENDED CGM (CGEM) STANDARDS EFFORTS

TABLE OF CONTENTS

1. Foreword	1
2. Summary and Conclusions	1
3. Recommendations	2
4. Activities in FY89	3
4.1 Motivation for CGEM	3
4.2 Historical Overview	3
4.3 Review of Goals for FY89	5
4.4 Key Activities of 1989	6
4.4.1 Addendum 1	6
4.4.1.1 Initial Status and Goals	6
4.4.1.2 Houston X3H3 Meeting	6
4.4.1.3 DAD Ballot & ANSI Letter Ballot	7
4.4.1.4 X3H3 Milpitas Meeting	7
4.4.1.5 Between Milpitas and Waikoloa	8
4.4.1.6 Waikoloa MRG Meeting	8
4.4.1.7 Nashua X3H3 Meeting	11
4.4.1.8 Status & Remaining Work	11
4.4.1.9 ANSI Processing of Addendum 1	12
4.4.2 Addendum 3	12
4.4.2.1 Initial Status and Goals	12
4.4.2.2 Redondo Beach Text Meeting	13
4.4.2.3 Munich Study Group Meeting	14
4.4.2.4 Milpitas X3H3 Meeting: Liaisons and Drafting	15
4.4.2.5 Between Munich and Waikoloa	15
4.4.2.6 Waikoloa Meeting	16
4.4.2.7 Post Waikoloa & ANSI Letter Ballot	17
4.4.2.8 Nashua X3H3 Meeting	17
4.4.2.9 Status & Remaining Work	18
4.4.3 Addendum 2	19
4.4.4 Coordination between CGM Addenda and Graphical Registration	20

TABLE OF CONTENTS (Continued)

Attachment 1:	US Vote and Comments on CGM Addendum 1 DAD Ballot	21
Attachment 2:	US Contribution to the Improved Graphical Text Model	31
Attachment 3:	Minutes of the Munich CGM Extension Meeting .	45
Attachment 4:	Addendum 3 New Work Item Proposal Addendum 3 Requirements Document	55
Attachment 5:	Final Report of CGM Extension Study Period .	63
Attachment 6:	Addendum 3 Working Draft	73
Attachment 7:	US Comments on Addendum 3 Working Draft . . .	115
Attachment 8:	Draft Addendum Text for CGM Addendum 1 . . .	127

1. Foreword

NIST/NCSL has in the past participated in the CGEM work through the CALS representative, Lofton Henderson, on the ANSI and ISO committees, and has continued this participation in FY89. In FY89, which ended on September 30, 1989, the work for this project had three principle aspects: the working meetings of the graphics and metafile experts of ANSI and ISO; inter-meeting work of preparing and coordinating position papers, baseline standards documents, and ballot responses; coordination and liaison with the Graphical Registration work.

This final report summarizes the progress that was made at the working meetings, the current status of CGM extensions work, projected timetables for completion, and recommendations for future work. This report also presents details of key domestic and international meetings for CGM extensions during FY89. Finally, key documents such as standards drafts, study reports, and US position papers which are a result of work under this contract are included as attachments.

2. Summary and Conclusions

Previous work has focussed on defining the CALS requirements for CGEM, getting those requirements endorsed by ANSI, and introducing those requirements into the ISO CGM addenda processing. The requirements definition and ANSI X3H3 endorsement were largely accomplished. Getting functionality into CGM Addendum 1 that meets some of the CALS needs has been largely accomplished. The major uncompleted work at the start of this fiscal year was getting ISO endorsement of the need for further extensions, the Addendum 3 project, and getting technical work underway on the project.

Specific goals for FY89 were:

1. Complete processing of CGM Addendum 1.
2. See Addendum 3 through the ISO NWI (New Work Item) process and expedite technical work on it.
3. Coordinate the CGEM work with registration proposals to insure maximum compatibility.

It appears that CGM Addendum 1 will complete soon, and it will have intact the capabilities that CALS has been working to put in and keep in. The final ISO balloting occurred early in the year,

and there were few significant changes as a result. This is as desired; NIST/NCSL wished the content to be stabilized and "frozen" and the processing wrapped up.

Most of the extensions identified by CALS requirements studies are addressed in Addendum 3. At the start of the fiscal year a formal ISO process was beginning, under the leadership of the NIST representative, to initiate Addendum 3 as an ISO project. This process went exactly on schedule, and with the desired results, and is just completing as this fiscal year completes. (NOTE: There is one last step, which has gotten delayed, but it is hoped that this is a formality.)

Slightly ahead of schedule a Working Draft for Addendum 3 has been produced, circulated for ISO comment, and the US has had a letter ballot and formulated extensive technical comments. Assuming adequate resource commitments and no serious dissension among the member nations, the project is on track. According to the schedule in the New Work Item, final text should be available in April 1991.

There is a potential problem looming, however. NIST/NCSL cannot be sure, despite the apparent ballot results, that the European members of SC24 share the same priorities and are willing to commit resources sufficient to complete the work, with the required content, in a reasonable timeframe. This should become apparent at the SC24 meeting in October 1989. ASC X3H3 will have to continue to have a fallback or contingency - possibly quick domestic processing as a US standard - if it is estimated that the project might bog down in ISO.

In addition to working on the CGM addenda, the NIST/NCSL representative reviewed and coordinated with the CALS Graphical Registration proposals during this fiscal year.

3. Recommendations

Further work is required if the Addendum 3 project is actually to complete expeditiously (or at all). It has been principally the NIST/NCSL representative who has been keeping the work alive and on track in both ANSI and ISO. Continued participation is essential.

NIST/NCSL will be monitoring and assessing the study work which is just commencing to define what, if any, 3D metafile requirements exist for effective support of the emerging product data standards (PDES and STEP in particular).

4. Activities in FY89

The remainder of this report contains details of the working meetings and progress on the CGM addenda during this fiscal year. Attachments containing key documents are included at the end. The material in the next two subsections is condensed from previous final reports on this subject and is repeated here for reference.

4.1 Motivation for CGEM

The CGM standard upon which MIL-D-28003 is based, ANS X3.122-1986, was completed in 1986. It is a "least common denominator" graphical file interchange standard. That is, it provides a suitable basic picture interchange format for diverse application areas. Its scope and content were not derived from any particular application area, but rather more from the content of other general purpose computer graphics standards.

This expedited the processing of the standard, at the cost of efficiency of usage in some application environments. The CALS application areas of technical illustration, technical publications, and compound document exchange comprise such environments. While experience shows that even the current CGM is more efficient than specifications such as page description languages (e.g., PostScript) and engineering data specifications (e.g., IGES) for graphical interchange, nevertheless CALS requirements studies show that the efficiency and fidelity of interchange could be improved with a well designed set of extensions to CGM.

For these reasons an extension process was commenced in 1986 within ISO SC24 to extend CGM functionality as required by its more advanced metafile application constituents.

4.2 Historical Overview

Two addenda were officially commenced in 1986:

- o Addendum 1: originally intended to support certain needs of GKS, and replace the non-standard specification in Annex E of GKS (i.e., provide a GKSM, workstation session capture metafile, to replace Annex E of GKS);
- o Addendum 2: originally intended to support the metafile requirements of GKS-3D, and replace the non-standard Annex E of GKS-3D.

Due in part to the successes of the work done in previous years for this task:

- o ASC X3H3 endorsed and supported the CGM extensions sought for CALS.
- o The scope of Addendum 1 was expanded to include some of these extensions: internal symbol libraries, additional geometric primitives, and basic raster primitives.
- o Addendum 3 was commenced, to extend CGM further as required by technical illustration, engineering drawing, and publishing environments.
- o Requirements statements and baseline technical documents for Addendum 3 were produced within ANSI and circulated within ISO as well. These materials were largely derived from CALS requirements studies and CALS activities.
- o A Metafile Reference Model was devised within ANSI, submitted to and accepted by ISO, with the result that the content and targets of the various addenda were significantly reorganized and redefined. Relatively uninteresting GKS-related content, which would have caused unnecessary and undesirable complication as a CGM addendum, was thereby split off and attached to GKS.

The impact of CALS participation in the ANSI and ISO work under this and previous contracts is significant. The NIST/NCSL representative was the document editor of CGM (both the ANSI and ISO documents), has led and continues to lead the CGM work within X3H3.3, has led and continues to lead the US delegation at ISO metafile meetings, and is the ISO leader of the Addendum 3 Study Period. Without this participation the CGEM work would have progressed much more slowly, and in fact Addendum 3 might not have happened at all.

At the start of FY89, Addendum 1 was well advanced in its processing and was expected to change little (see the detailed activity reports following, however). Addendum 3 was entering a study phase in ISO, during which the requirements were to be studied and agreed upon. This work was largely based on US work in the previous year.

The following list summarizes the requirements which were identified and designated as the scope of Addendum 3 at the start of this Addendum 3 Study Period, which corresponded roughly with the start of this fiscal year.

1. Internal symbol libraries;

2. Reference to and invocation of pre-defined external symbol libraries;
3. Advanced drawing capabilities, including:
 - o user defined line type;
 - o user defined hatch style;
 - o a number of additional line types;
 - o a number of additional hatch styles;
 - o several types of spline curves;
 - o conics and conic arcs;
 - o closed figure primitive;
 - o arbitrary clipping boundary;
4. A number and variety of fonts;
5. A completely new text model based on the work of ISO 9541, Information Processing--Font and Character Information Exchange;
6. Additional raster primitives (and associated attributes for image processing);

4.3 Review of Goals for FY89

The goals for the FY89 CALS CGEM project were:

1. Addendum 3: completion of the formal New Work Item (NWI) procedures for Addendum 3 within ISO and resumption of the technical work, including:
 - o endorsement of the need and content of CGM Addendum 3 by the CGM Extensions Study Period;
 - o production of the necessary NWI and supporting Requirements Document;
 - o initiation of SC24 NWI ballot and processing of results;
 - o technical work on the initial draft of the functional content, which was a US contribution that was a year or so old.
2. Addendum 1: Finish it by: circulation of Addendum 1 DAD (Draft Addendum) text in X3H3 and generation of a US position on the SC24 DAD ballot; processing the SC24 ballot results and producing final text.

3. Addendum 2 (3D): Prevent the project from draining resources from the Addendum 1 and Addendum 3 work until the project gets a well-defined and useful scope. This may in fact happen now as a consequence of liaison with product data standards committees within ISO. A 3D graphical file format may be defined to serve the needs of STEP/PDES. This would be a development of interest to CALS. However, the current Addendum 2 proposals are relatively unfocused and uninteresting.

4.4 Key Activities of 1989

These subsections, describing the key meetings and other activities during the task period, are organized first by addendum and then chronologically within each addendum section.

4.4.1 Addendum 1

4.4.1.1 Initial Status and Goals

At the close of work in FY88 the Addendum 1 project had been split into two pieces: a static picture capture Addendum 1 to the CGM standard; and a dynamic audit trail metafile Addendum 1 to the GKS standard. This was in consequence of Metafile Reference Model work developed within X3H3.3/CGM (the US metafile group) under the leadership of the NIST/NCSL representative, and carried to the ISO SC24 meeting (Tucson, June 1988). The split separated the parts of interest to CALS - CGM Addendum 1 - from functionality to provide a GKSM capability for GKS (which is of no interest to CALS). At the same ISO meeting it was also agreed to skip a second round of PDAD (Proposed Draft Addendum) ballot and go straight to DAD status (DAD is the final processing stage for an ISO addendum). This was largely due to the position of the US delegation, which was being lead by the NIST/NCSL representative, and it saved a potential 9 month delay in completion of CGM Addendum 1.

At the beginning of work in FY 1989 then, the DAD text of CGM Addendum 1 was being circulated for a 6-month ISO SC24 ballot. With proper handling of voting and processing this ballot would be the last processing step for CGM Addendum 1 and its completion would be expected in late 1989 or early 1990.

4.4.1.2 Houston X3H3 Meeting

There was a meeting of the ANSI X3H3 committee in Houston, 17-21 October 1988. There was no activity of interest to CALS on the

CGM Addendum 1 project at this meeting. The DAD text had not yet been received from ISO. There was only announcement and planning for an X3H3 letter ballot on the DAD text when it arrived in late 1988.

4.4.1.3 DAD Ballot & ANSI Letter Ballot

The DAD text was received by the NIST/NCSL representative in December 1988 directly from the document editor in England. An X3H3 letter ballot was prepared and mailed with the CGM DAD.1 text for a 30-day ballot, to collect and formulate the US position for the SC24 DAD ballot. The ballot period was 6 January 1989 to 6 February 1989. The DAD.1 document is similar to the text in Attachment 8 (Attachment 8 is the draft final text for CGM Addendum 1, which is the result of applying the DAD.1 ballot results to the DAD.1 text). Note that the GKS Addendum 1 document was being balloted simultaneously, but has not been included as it is of little interest to CALS.

4.4.1.4 X3H3 Milpitas Meeting

There was a scheduled X3H3 meeting in Milpitas, CA, February 6-10, 1989. The agenda for CGM work was mostly devoted to processing the comments from the DAD.1 letter ballot. The goal of NIST/NCSL for the US position was two-fold: there should be absolute minimal change in the DAD.1 text, to avoid forcing a second DAD ballot; and the CALS content of Addendum 1 should stay intact. A second DAD ballot at the ISO level would cause 9-12 months delay in the completion of processing. The ISO rules as traditionally interpreted by SC24 would require a second DAD ballot if there were any technical change.

One point in the US position deserves explanation. The US voted to remove PIXEL ARRAY from Addendum 1. This function (as reported in 1987) was one of the CALS functions and was supported by CALS requirements studies. The NIST/NCSL representative supported the removal because the function had been formulated in a way which did not meet CALS requirements. The function was taken straight from the CGI (Computer Graphics Interface) standard, and had a parameterization which was device dependent, could not be coordinated with the other geometric primitives of CGM, and was conceptually different from the PEL ARRAY GDP which had been promoted for Graphical Registration.

This technical deficiency alone was sufficient reason for removal of PIXEL ARRAY. Opposition and negative votes were anticipated from at least UK and France as well.

Attachment 1 contains the US vote and comments on the ISO ballot, which was formulated at the Milpitas meeting under the leadership of the NIST/NCSL representative. Note that the PIXEL ARRAY comment urges that it be reformulated and properly included in Addendum 3. Note also that the US voted "no". This is procedurally required by ANSI if a technical (as opposed to editorial) comment is being made. As delegation leader to the ISO editing meeting on Addendum 1, the NIST/NCSL representative was empowered to change the US vote to "yes" if the US comments were adequately addressed at the meeting.

The ISO DAD ballot was scheduled to end 15 June 1989, and results were scheduled to be processed at an SC24/WG3 MRG (Metafile Rapporteur Group, the sub-group of WG3 responsible for CGM maintenance and CGM extensions) editing meeting 20-25 June 1989 in Waikoloa, HI.

4.4.1.5 Between Milpitas and Waikoloa

Private communication by electronic mail took place between the NIST/NCSL representative, who is the principle US metafile delegate to the SC24/WG3 MRG, and the principle metafile delegates of the UK and Germany from February through June. It became clear that there were a number of "no" votes on the CGM DAD.1 text, and there were a number of technical issues. This raised the possibility that technical changes would force another DAD ballot. The UK delegate thought it inevitable. The US thought it must be avoided at all costs, in part because the SC24 metafile workers had to be freed up to work on Addendum 3. Germany agreed with the US. The UK finally agreed that the US could make technical changes and avoid a second DAD ballot if there were consensus on all of the changes as per ISO rules. For this reason the US circulated and debated the various national positions quite a bit during the months before Waikoloa, and were in substantial agreement before the meeting commenced.

4.4.1.6 Waikoloa MRG Meeting

The MRG met in Waikoloa, HI, 20-25 June to process the results of the DAD ballot. The US delegation consisted of one person, the NIST/NCSL representative. The goal was to resolve all negative comments and get unanimous approval, amend the document as agreed, and thereby finish work on CGM Addendum 1.

The initial count on the DAD ballot was: 8 approve, 5 disapprove. Disapproving were Austria, France, Germany, UK, and US, in other words all the key countries. The minutes of the meeting are not yet available, so cannot be included with this report. A summary is presented below.

The negative comments fell into a few categories:

- o Unhappiness with PIXEL ARRAY (nearly unanimous);
- o Divergence between the specifications of Addendum 1 and CGI where they overlapped (much of the functionality of Addendum 1 was borrowed from CGI and the two standards were to remain identical);
- o Misunderstanding of the meaning of some of the items in Addendum 1 (a number of French comments fell into this category).

The biggest problems during the six days of issues reconciliation and editorial work came from attempting to keep Addendum 1 and CGI (and CGM itself!) identical where they overlap. CALS does not consider CGI an important standard at this time; however, in the original model of the relationship between the standards CGM is basically identical to the output functions of CGI. Many still believe that this is a critical principle, and that abandoning it will impose unnecessary burdens on the computer graphics industry and other industries which rely on graphics standards. So effort is always expended to keep the two together. However, there seem to be those working on CGI who have forgotten this principle or no longer believe in it, and this creates conflicts such as occurred at the Waikoloa editing meeting. The CGI committee was meeting in parallel to process the results of its 2nd DP ballot and attempt to advance it to DIS stage.

The worst of the compatibility problems came when it was realized that CGI had changed the method by which clipping behaved under the Copy Segment function and its transformation. This issue is important to CALS, because one of the main pieces of functionality of interest to CALS is the Global Segment feature which was built into Addendum 1. In some cases (the default case in fact) the new CGI model had a tedious and complex scheme whereby clipping rectangles could accumulate and become arbitrary convex polygons. This is not the way CALS wants to use Global Segments. The CGM committee was unanimous in not wanting to have this functionality. After much debate with the CGI committee, the best compromise that could be achieved was: repackaging the functionality in such a way that an application profile can easily exclude the features that it does not want.

As a result of the editing meeting, all negatives but Austria's were resolved on CGM Addendum 1. Germany, UK, and the US participated in the editing meeting, with occasional input from France and Austria. The remaining negative from Austria was based on a single technical objection.

The editing meeting produced markup for complete final text, and the document editor (Anne Mumford of UK) will have final text around late August.

France wanted a second DAD ballot because of the changes that were made. The US, with support of UK and Germany, proposed a methodology that is somewhat unconventional for SC24 but would save another ballot round. The five nations represented at the CGM meeting would agree that the Addendum was acceptable with those modifications agreed to at the meeting, and the document editor would circulate the document for a six week review to verify that the agreed changes were properly implemented. After that review the document would go to ISO Central Secretariat with the recommendation that it be made an international standard. It is hoped that the Secretariat will progress the document, but this is not certain because there is an outstanding negative (Austria).

Except for PIXEL ARRAY, no significant functionality has been added to or deleted from Addendum 1, but some existing functionality was repackaged. The processing of comments resulted in the following changes to CGM Addendum 1:

1. PIXEL ARRAY is removed; improved raster capabilities will be revisited in Addendum 3 and treated in a coherent manner.
2. METAFILE CATEGORY is removed; the current basic static category is preserved by allowing "VERSION 1" to appear in the Metafile Descriptor. Annex A is corrected and made into a new normative Annex J (to define VERSION 1).
3. In a repackaging of functionality, clipping is removed from INHERITANCE FILTER; the exact same clipping features are now selected by a new function CLIPPING INHERITANCE.
4. The IMPLICIT EDGE VISIBILITY (a feature of Closed Figures) was discovered not to work properly, so was removed and its functionality replaced by CONNECTING EDGE.
5. A new datatype NAME, with its own NAME PRECISION, will replace datatypes ASN, PN, and SN.

Although GKS Addendum 1 (the other half of the original CGM Addendum 1 after the split at Tucson) is not of interest to CALS, there was one result on its processing which is related to the CGM Addendum work. The UK took the position that a standard GKSM is not needed, and that its standardization will confuse the marketplace. They wanted the project dropped altogether. Although sympathetic, the US would not vote this position because of agreements made earlier with GKSM supporters to support the work (in return for them not holding up CGM).

Instead changes were made to GKS Addendum 1 to make it conform much more closely to the GKS notion of a session-capture metafile, and to make it much more distinct from CGM (and hopefully less likely to be confused with CGM in the real world). The delimiters of the GKSM (GKS Addendum 1) were renamed, to reduce the chances of metafile interpreters confusing a GKSM with a CGM. In particular, the BEGIN METAFILE element was renamed BEGIN GKS SESSION METAFILE, and given a unique encoding. Similar changes were made for BEGIN PICTURE and END METAFILE. BEGIN PICTURE BODY and END PICTURE were removed. Thus in any encoding the GKSM becomes structurally different and is introduced by different delimiters.

One final note of interest for the Waikoloa meeting: this was the last meeting for Eckhard Moeller, of Germany, the rapporteur of the WG3 Metafile Rapporteur Group. Anne Mumford of the UK assumes the rapporteur position starting at the Brazil meeting. The NIST/NCSL representative could probably have had this position. However, chairing the group compromises the chair's ability to influence the technical content of the work. The latter role is more important for CALS, on the Addendum 3 work, on which the MRG should henceforth be spending most of its effort.

4.4.1.7 Nashua X3H3 Meeting

An X3H3 meeting took place in Nashua, NH during the week of September 25-29, 1989. There was no significant activity on Addendum 1 at this meeting.

4.4.1.8 Status & Remaining Work

The revised CGM Addendum 1 document, the draft final text, was received directly from the document editor in England in mid-September. Some 20 copies have been distributed to US experts for final review and accuracy checking.

It is clear that some parts of the text have not been completed as evidenced by the document editor's cover memo. In addition, a new annex containing a formal grammar was overlooked altogether. For these reasons, the text may have to be circulated again after these parts are supplied. If this is not done, then final text will not be reviewed by anyone except the document editor. This decision to circulate will be made at the SC24/WG3 MRG meeting in October. If another round of review is required, the final acceptance of the text will be delayed by another 2 months.

4.4.1.9 ANSI Processing of Addendum 1

As shown in the preceding sections, all work that has occurred is ISO work towards an Addendum to the ISO version of CGM (ISO 8632), with all US activity contributing to that work. MIL-D-28003 is based on FIPS PUB 128, which is identical to the US version of CGM (ANSI X3.122). The US and ISO versions of CGM are in fact identical in content, but differ in editorial style.

An addendum (see Attachment 8) is an editing script. As such, it is not a complete standalone document but consists of change directives against the base document. To adopt Addendum 1 in the US would require a document editor to revise the ISO text.

Consequently, ANSI has devised new procedures which make it much easier to adopt ISO work automatically as ANSI standards, and to designate ISO standards as ANSI standards. The NIST/NCSL representative is seeking such a change on status for ANSI CGM. Thus, the current ANSI document would be withdrawn and replaced by the ISO document. The ANSI and ISO versions of CGM would then be the same document. After this change, ISO Addendum 1 could be adopted directly. The implications on MIL-D-28003 should be minor, since it would only have to be checked to make sure that page references are correct, and that the proper base document is referenced.

4.4.2 Addendum 3

4.4.2.1 Initial Status and Goals

At the start of FY89, the US had been working on Addendum 3 project for a year. In prior CALS work, the metafile requirements of CALS which were not met by CGM and not met by Addendum 1 (or Addendum 2) were identified. An Addendum 3 was proposed, and its approximate contents were defined. A strategic choice was faced: should this be expedited as an ANSI project, or should it be pursued as an ISO project? Although the former would probably be more expeditious, past experience has shown that if ISO undertook such work, and if it did so before ANSI completed, then it would be likely that the ANSI work would be put aside and ANSI would join the ISO effort. For this reason, it was decided to pursue the work aggressively in ISO.

The US first proposed the project at the initial SC24 meeting in December 1987. The project was not accepted at that time,

because SC24 procedures and priorities were in flux, and there was some dispute within ISO about which committee should lead on the Addendum 3 content. Consideration of the project was deferred to a special study meeting of SC24 procedures which took place in April 1988. Primarily due to lack of time, the meeting did not address Addendum 3, but the NIST/NCSL representative did get the point across that the US would undertake the project on its own if ISO did not move on it. At the SC24 plenary in June 1988 a study group was established, lead by the NIST/NCSL representative, to initiate the project under new SC24 procedures, get consensus on its scope, and coordinate with parallel study groups for Improved Graphical Text Model and for Product Data Geometry (PDG).

The parallel study groups were formed because their two technology areas would be important for all new graphics standards. The text model of all first generation standards (CGM, GKS, PHIGS, CGI) was acknowledged to be inadequate for modern typographic, engineering, and presentation requirements. The purpose of the text study group was to identify a new common model for the second generation of standards, which would include Addendum 3.

The PDG group was formed for a similar reason. The set of geometric primitives in most first generation standards was rudimentary. CGM was better, including some conics, etc. But modern practice requires a more substantial set, including such primitives as various sorts of splines and full conics. The PDG group was charged with defining the relationship between product data standards (PDES, STEP, etc) and graphics standards. From this relationship might be deduced further geometric primitives which should be in the graphics standards in order to fully support product data standards.

The Addendum 3 study group was to report at the next SC24 plenary, October 1989, and was to complete all of the required New Work Item (NWI) processing steps and produce a Working Draft (WD) Addendum 3 by then. The initial meeting of the study group was scheduled at the beginning of FY89, and was arranged so that the two technology groups (Text and PDG) would meet immediately before the Addendum 3 group at the same location.

4.1.2.2 Redondo Beach Text Meeting

An ad hoc meeting of several US experts, including the NIST/NCSL representative, was held December 1-2, 1988 in Redondo Beach, CA. The purpose of the meeting was to write a position paper for the upcoming meeting of the text study group. This paper listed all the features that an improved text model should have. It may be found in Attachment 2 of this report.

4.4.2.3 Munich Study Group Meeting

The first set of study group meetings was scheduled for January 16-20, 1989 in Munich, West Germany. Text and geometry groups were to meet in parallel during the first half of the week, and CGM extensions (Addendum 3) during the second half. Thus, metafile experts would be able to participate in one of the technology meetings, and the technology specialists would be able to participate in the CGM extensions meeting. The meetings were held as scheduled, despite last minute attempts of PDG to move its meeting to a later date. The NIST/NCSL representative, who is the leader of the CGM extensions group, was unable to attend due to a family emergency. Peter Bono, chair of X3H3, chaired the meeting in his absence.

US input to the CGM extensions meeting consisted of a draft New Work Item (NWI) proposal and supporting material for producing a Requirements Document (these were listed as base documents in the meeting call, which was written and submitted by the NIST/NCSL representative). The supporting requirements material included previous CALS requirements studies and parts of earlier Addendum 3 proposals.

The meeting resulted in the group endorsing the need for Addendum 3, and advising that it be expedited. A draft NWI and Requirements Document were produced as required by new SC24 procedures. These were substantially the same as the input documents - all of the important CALS requirements were endorsed and not many more were added. Details can be found in the attachments.

Several milestones in the NWI schedule should be noted:

1. SC24 ballot on the NWI and Requirements to complete by the June 1989 Waikoloa meeting;
2. processing results at that meeting;
3. JTC1 ballot to complete by the October 1989 SC24 plenary;
4. substantially complete Working Draft of Addendum 3 to be done by the end of that meeting;
5. final text for Addendum 3 in April 1991.

The minutes of the Munich meeting are found in Attachment 3 of this report. Attachment 4 contains the final NWI and

Requirements Document for the project. These were actually produced at the Waikoloa meeting in June 89, but included here because they are so similar to the drafts from the Munich meeting that the difference is negligible (there are a couple additions to the bibliography and a couple additions to the list of liaison projects).

The output of the meeting also included three 1-page letters, to the rapporteurs of the SC24/WG1 Reference Models group, Requirements group, and the rapporteur of SC24/WG1 himself. These letters basically asked the recipients to carry out the NWI processing steps required under the new SC24 procedures.

4.4.2.4 Milpitas X3H3 Meeting: Liaisons and Drafting

There was a scheduled X3H3 meeting in Milpitas, CA, February 6-10, 1989. The agenda for CGM work was mostly devoted to processing the comments from the CGM Addendum 1 letter ballot (see the above section on Addendum 1). There was little time for Addendum 3 work. However:

- o there were extensive liaison meetings between CGM people and experts from other technology areas: text, CGI, geometry, image storage, etc.
- o the NIST/NCSL representative leading the CGM meetings made interim writing assignments to the X3H3.3 CGM people with the goal of assembling a new Addendum 3 baseline document prior to the Waikoloa meeting.

4.4.2.5 Between Munich and Waikoloa

Many of the details of processing the Addendum 3 NWI during this period are found in the final report of the rapporteur of the CGM Extensions Study Period (the NIST/NCSL representative), which is in Attachment 5. To summarize briefly:

1. As per the new SC24 procedures, the NWI and Requirements Document went to the SC24/WG1 Reference Model meeting the week following the Munich meeting. No changes were requested by that group.
2. As per the new SC24 procedures, the NWI and Requirements Document went to the SC24/WG1 Requirements rapporteur for their late February meeting. That meeting was cancelled and so no changes were requested by that group.
3. The SC24/WG1 group at its February meeting endorsed the Munich results and asked the SC24 Secretariat to circulate same immediately for a SC24 NWI ballot.

4. This was done, and the ballot closed June 15, 1989.

In order to form the US position on this ballot, the NIST/NCSL representative prepared an X3H3 letter ballot. This was sent to the X3H3 mailer in early April for a 30-day ballot period. Mailing problems with that X3H3 officer resulted in the ballot being delayed until early June. This was too late, so the ballot was scrapped and the US voted "yes without comment" on the NWI.

In this period, the NIST/NCSL representative had to prepare and circulate the call for the next CGM Extensions Study Period (Addendum 3 study group) meeting. SC24 procedures require that this be done at least 2 months in advance of the meeting, which was scheduled for Waikoloa in late June.

Finally in this period, the NIST/NCSL representative coordinated the assembly of the Addendum 3 draft by X3H3.3 CGM people. In early June this draft was mailed to the international attendees of the Waikoloa meeting.

4.4.2.6 Waikoloa Meeting

The final scheduled meeting of the CGM Extensions Study Period took place in Waikoloa, HI, June 26-28, 1989. This was immediately following the CGM Addendum 1 editing meeting. The meeting was chaired by the NIST/NCSL representative. The goals were:

- o process the results of the NWI ballot;
- o revise the NWI and requirements if necessary and send to the SC24 Secretariat for immediate JTC1 ballot (JTC1 is the parent organization of SC24);
- o continue technical work on the baseline document;
- o begin work on the final report of the study period.

All of these goals were met. The study period final report in Attachment 5 contains details. This final report was actually produced subsequent to the meeting.

One point in the final report deserves note. It is clear that SC24/WG3 MRG is facing a resource crunch. After all of this work to get the project through the ISO procedures, the question of

whether ISO has the resources to deliver the project on schedule must be seriously looked at. This will be determined at the October meeting. The US will need to develop a contingency strategy in case ISO cannot deliver.

4.4.2.7 Post Waikoloa & ANSI Letter Ballot

The work done at Waikoloa on the Working Draft of Addendum 3 was incorporated by the document editor. This document is contained in Attachment 6. In late July, it was sent to the SC24 Secretariat for SC24 circulation and comment. This is normally a 3-month period, but early comment had been requested so that technical work may take place at the SC24 plenary 15-30 October 1989. An X3H3 letter ballot had been drafted and circulated as well. The purpose was to solicit comments on the Working Draft of Addendum 3 and use them during the X3H3 meeting of September 25-29, 1989 in formulating the US position.

It has since been learned, that the JTC1 ballot period had not even commenced, although the Addendum 3 schedule showed it being complete by the SC24 meeting in October. The reason for this delay is disorganization within the JTC1 Secretariat and lack of communication between it and the SC24 Secretariat. The consequence is that Addendum 3 is still not an official ISO project. Although JTC1 approval is usually considered a formality it is nonetheless the last required step in the process.

It is also at the JTC1 voting stage that the member nations formally indicate their levels of participation and of resources. This will be a key issue to the success of the project in SC24.

4.4.2.8 Nashua X3H3 Meeting

The X3H3.3 CGM subgroup had 6 attendees at the Nashua meeting, 25-29 September. With the exception of a two-hour liaison meeting with 3D and product data experts to define a response to an outstanding Addendum 2 ballot, the entire meeting was devoted to processing the substantial X3H3 comments on the Addendum 3 Working Draft (WD) and forming a US position for the ISO comment period.

The US position is contained in Attachment 7. In summary, the WD has significant omissions and in other respects is very rough. This is to be expected at this stage in the standards process. The WD will require significant improvement, however, before the first official voting can commence in ISO. This is the PDAD ballot. It is possible that these improvements can be made at the SC24/WG3 meeting in October.

The US position also makes clear what the priorities of the work should be. Private communications with other national delegates indicate the possibility of other prioritizations which may seriously slow the project (see next section).

The key technical problems can be briefly summarized:

1. the geometric primitives are incomplete with respect to what was specified in the NWI and Requirements documents;
2. the formulation of some of the geometric primitives, as well as the image primitives, needs improvement;
3. numerous technical and editorial details concerning the additional attributes, color models, etc need to be corrected;
4. the improved text capabilities need a thorough overhaul, including: clarification of how external font resources are accessed; better glyph access methods; clarification and improvement of font callout and substitution techniques; removal of glyph definition primitives from the metafile.

4.4.2.9 Status & Remaining Work

It appears that the Addendum 3 project has been accepted by ISO, but the final round of balloting in JTC1 has not yet taken place. There is a procedural possibility that this can be skipped, that SC24 can issue a resolution that this is a simple revision of an existing standard and so JTC1 (SC24's parent body) need not vote on it. This will have to be explored at the October SC24 meeting.

Although the formal acceptance seems likely, there is still danger that the project will not progress adequately. This comes from views of some of the other national delegations about metafile extensions. The US view is that CGM relates to other computer graphics standards, but relates more to "current practice" in engineering, publishing, and graphics arts. This means that users in these areas are retro-fitting CGM import and export filters to existing proprietary products. In the process, they care little about the other work of SC24; they just want to exchange graphical information between heterogeneous systems. In other words, in the view of US users, Addendum 3 has more to do with IGES, PDES and PostScript than it does with GKS.

The view in Europe tends to be less pragmatic and more academic. There is high interest in Europe in a new Application Programmer Interface standard (API) - a revision to or replacement of GKS. Comments have already been heard and seen to the effect that the Addendum 3 work must go no faster than the new API work, that they must work out access to all new technology in compatible (meaning identical, to the commenters being referred to) ways. These sentiments seem to echo similar conflicts that occurred during CGM standardization, and which slowed the standard down by 1-2 years.

There are also indications that some think the scarce resources of SC24/WG3 should be split between Addendum 3 and Addendum 2 (3D). NIST/NCSL believes there are barely sufficient resources to do the Addendum 3 work. Any attempt to carry on Addendum 2 as well, without additional staff, would have serious impact.

The US may thus face an important and strategic decision in the October SC24 meeting: should it continue to support the Addendum 3 work as an ISO project; or should it attempt to block further ISO work on metafiles for a couple of years and go back to ANSI processing? The latter is clearly a drastic resort. It has been the US position that the work belonged in ISO. ISO endorsed this, and the tentative schedule, in the NWI process. US companies definitely will fare better in international markets if their standards are international standards. However, if the other ISO member nations do not actually have the will to follow through with the commitment to constituency and schedule that was indicated in the NWI process, then the US must consider withdrawing the project.

4.4.3 Addendum 2

Since the Waikoloa meeting a version of the Addendum 2 (the 3D addendum) document has been circulated for PDAD ballot within SC24. The US had to respond to this ballot. So a second X3H3 letter ballot was put together and circulated. This was processed at the September meeting in Nashua.

The US position was defined at the Nashua meeting in liaison with PHIGS experts and those interested in product data standards (STEP/PDES). It appears that the PHIGS extension project in X3H3 will take interest in this project, as a means of providing support for STEP, and will help to define a useful scope. There is the possibility of additional staff and help from the 3D subgroups, in which case there could be sufficient staff to process both Addendum 3 (which is important to CALS), and Addendum 2 (which is not).

These points are speculative, however, as the necessary preliminary liaison between the 3D experts and the product data standards committees has not yet occurred. Accordingly the US position on the Addendum 2 PDAD ballot contained four simple points: no further effort should be expended on the Addendum 2 as currently specified; the need for a 3D metafile should be defined by 3D experts and metafile experts in liaison with TC184 (the ISO committee working on STEP); any resulting metafile project should be jointly executed by 3D and metafile experts; WG3 must have additional resources to execute its part of any resulting project. Should a 3D metafile project result, it is anticipated that the 3D experts would principally define its functional and semantic content, as well as its position in the reference model, and metafile experts would design the metafile and write its encodings.

4.4.4 Coordination between CGM Addenda and Graphical Registration

NIST/NCSL has been sponsoring registration of graphical items for CALS. These are intended to provide a short term solution to functions needed by CALS that are being pursued through the slower formal standards process (Addendum 1 and Addendum 3).

Because these are addressing the same needs, the formulations in Graphical Registration and the addenda should usually be very similar (there are cases where the different mechanism of the GDP and ESCAPE elements which are registered justify some difference in formulation).

During this fiscal year there has been frequent liaison between the NIST/NCSL representative and NIST/NCSL to coordinate the content of CGM Addendum 3 with the registration proposals. The results have been adjustments to proposals in Registration Batch 2 and Registration Batch 3 (see the final report titled FINAL REPORT, CALS FY89 SOW TASK 4.3.2, MIL-D-28003 REVISION RECOMMENDATIONS) as well as reformulation of Working Draft Addendum 3 and additional specifications for the pending revision of MIL-D-28003. The effect of the adjustments is generally a convergence of the proposals and the draft addenda.

Attachment 1:

US Vote and Comments on CGM Addendum 1 DAD Ballot

U.S. Comments on ISO 8632-1/DAD.1 (CGM Add.1)

The U.S. disapproves ISO 8632-1/DAD.1 with the following technical comment:

PIXEL ARRAY should be removed from CGM Addendum 1 and should be considered as part of the Addendum 3 project. There are several reasons: 1) Image storage and transfer capability is being studied for Addendum 3 and the PIXEL ARRAY capability should be included in this more comprehensive study; 2) The current PIXEL ARRAY formulation is based on the CGI formulation and the latter is considered unstable at this point; 3) The current formulation of PIXEL ARRAY is device dependent and apparently does not exist in the reference model at the same level as other CGM elements; its relationship to the other elements at least needs to be more carefully defined before being included in CGM extensions.

In addition, we note the following inconsistency with the 2nd DP text of CGI and request that this inconsistency be addressed jointly by the CGM and CGI RGs of WG3:

The behavior of CLIP RECTANGLE under COPY transformation differs between the CGI and CGM. We believe the CGM specification is more compatible with API standards. We understand that the CGI specification is still subject to change in this area. In any case, this must be resolved between CGI and CGM.

Editorial Comments:

- E1:** The discussion of the effects of anisotropic transformation in 4.12.4.5 has been clarified in the CGI. CGM should adopt the clarified wording.
- E2:** Section 4.12.4.4 should point out that SEGMENT PICK PRIORITY has no graphical effect and is available for application dependent communication between interpreters and generators. The same should be pointed out for PICK IDENTIFIER in section 4.7.9.
- E3:** In Sections 4.12.5 and 5.10.1.2, PICK IDENTIFIER and ASF's have been omitted from the description of INHERITANCE FILTER. CGM is intended to be the same as CGI in this area.
- E4:** Page 1, sub-clause 0.3, item c) should be deleted. It is not possible to anticipate what future standards will require. In any case, it does not add any useful information to the standard.
- E5:** Page 41, Table 3.1. The entries for SCALING MODE are wrong. The entire table should be carefully checked for correctness.
- E6:** Pages 56-57, sub-clauses 5.4.2-5.4.5. Reword these three sub-clauses along the lines of, "The COLOUR SELECTION MODE may be changed only within the picture description in category "basic-static". It may be allowed in the picture body in some of the other categories.

- E7: Page 58, sub-clause 5.4.6. This sentence is not useful and should be deleted.
- E8: New sub-clause 5.5.15. There seems to be a remnant of an old version (stacked attribute sets) indicated here. If attribute sets are named, is it not the case that the NAMED attribute set be the one restored, not the LAST one?
- E9: Page 103, new element defaults. SEGMENT DISPLAY PRIORITY and SEGMENT PICK PRIORITY should not be encoding dependent, but rather the default should be zero.
- E10: The grammar has not been carefully reviewed in the past and we request that a careful examination is done before IS text is produced.
- E11: For Note 1 under H 6.7, "action required" flag and "no-action" do not seem to be referenced anywhere else in the document.
- E12: There is an inconsistency in the usage of phrases "view surface" and "display surface". To be consistent with itself and with CGI, the phrase "drawing surface" might be a better choice.
- E13: All references to a CGM "function" should be replaced by "element".
- E14: In the definition of anisotropic mapping, we suggest replacing "to physical device units" with "distance units on the physical drawing surface".
- E15: In the definition of edge, we suggest replacing "The rendering of the boundary" with "The rendering of the perimeter" to avoid confusion about interior style HOLLOW (rendering of the boundary) versus EMPTY with edges visible (rendering of the edges or perimeter).
- E16: In the definition of isotropic mapping, we suggest replacing "device coordinates" with "distance on the drawing surface".
- E17: In the definition of size specification mode, we suggest replacing "the state list" with "the Modal State List". Use of the terms "state list" and "current state list" need to be looked at. The concept of "Modal State List" was introduced in clause 4.12.2.4. More needs to be said about this concept earlier so that it can be used and referred to where needed. Also, more could be said about the general states of the metafile interpreter. Table 3.1 is great, but there needs to be some more general discussion.
- E18: The definition of graphic object should be added to the definitions. It is used in many places, e.g. the object clipping mode concepts. Also, we suggest using only the term "graphic object" and not variations such as "graphical object".
- E19: In clause 5.4.9, the parameter should be called "device viewport specification mode".
- E20: Page 42, sub-clause 5.1 Abbreviations. The meaning of PN should be:
 PN Pick Name Pick Identifier
 Realization is an integer.
 Range is implementation dependent.
- E21: Page 12, sub-clause 4.2 - The reference should be to sub-clause 4.4.2.

E22: Page 2, 6th paragraph, "compound text": Change "text that contains" to "text that may contain" (a compound text string need not contain attribute changes, it's compound if it was specified with multiple text/append text elements).

E23: Page 4, Section 4.3.2.5: This should be called something other than "gksm" now. We suggest "Add.1-static-gks." GKSM should be reserved for the audit trail that is described in the other (GKS) addendum.

E24: Page 6, 4.3.4.3, end of 4.4 paragraph: It is very confusing, perhaps inappropriate, to keep referring to categories which are not static picture-capture metafiles. There aren't any defined in this addendum. Standards should not be written in such a way that they imply or assume very much about future extensions. There should be a single paragraph that says "Various restrictions (such as where elements are permitted) are permitted to be different in categories to be defined in future extensions, or in metafiles defined in other standards which are based on this one," and leave it at that.

E25: Page 7, Section 4.4.7, end of 3rd paragraph: "relative to the non-inverted viewport" does not convey the necessary information. 5.4.10 has the proper wording and it should be repeated here or referenced here.

E26: Page 8, paragraph 7: Under LOCUS THEN SHAPE, it should also be noted that a thick line whose locus is outside of the clip window will not have any portion visible even if its line width would carry some portion of the rendering into the clip rectangle (same as LOCUS clipping).

E27: Page 8, Section 4.5.2, paragraph 8: "When a width or size specification mode is 'scaled', the rendering of shape proceeds in DC space after the VDC-to-Device Mapping." It is unclear whether this simply applies to the anisotropic_mapping and wide_lines question, or whether this is implying that SHAPE CLIPPING doesn't work with scaled specification modes. Without using CGI's pipeline, much of the wording is unclear. SHAPE CLIPPING clips the same regardless of the specification mode (that the whole point), and the wording simply needs to be clarified.

E28: Page 15, Section 4.12.4.5, 4th paragraph from the bottom, last sentence: Since the segment transformation is VDC->VDC, the VCD->DC mapping (set up by VDC EXTENT and DEVICE VIEWPORT etc.) is applied afterwards. The last sentence of this paragraph could be read as meaning that the latter transformation is only applied if the SEGMENT TRANSFORMATION was not applied. We think the word "only" is needed after the word "using."

E29: Section 4.3.2.4 and 4.3.2.5 refer to "GKS". Please change this to the "IS #-year" form of reference.

E30: 4.12.3.1. "Each segment has a unique identifier." This is not exactly what is intended. We advocate "No two global segments may have the same identifier and no local segments may have an identifier the same as other local segment in the same picture or the same as a global segment."

E31: 4.12.3.2. This clause does not actually state how a segment is opened. More generally, it seems sloppy to use the GKS words of "OPEN" and "CLOSED" to refer to static picture capture CGM files. It seems more natural to use terms like "elements delimited by a BEGIN SEGMENT element and an END SEGMENT element" when talking about CGMs.

E32: Defining a local segment in a picture automatically includes that segment in the picture's image. This needs clarification.

E33: What are the highlighting, pick priority, and display priority for primitives outside segments? This needs clarification.

E34: 5.10.1.2, 6th line. To what does "(see below)" refer?

E35: Under "Page 6 clause 3" object clipping mode needs to specify what "LOCUS" and "SHAPE" clipping imply and how they differ. "LOCUS THEN SHAPE" appears to be the logical concatenation of the other two modes. Also, the definition of "global segments" should read "these are segments which..."

E36: Page 10, sub-clause 4.3 Note. Use "within the definition of a global segment" rather than the present "when" construction.

E37: Last addition to Page 10, sub-clause 4.3: Finish first sentence with "in a metafile of any category other than..."

E38: Clause 4.3.4.1 should re-iterate that only a metafile of category "basic-static" is permitted to omit the metafile category element as implied by the first paragraph of Addendum 1, page 3 (sub-clause 4.3).

E39: Clause 4.4 first paragraph: Strike the last sentence or rephrase. All IS 8632 metafiles (regardless of Addenda work) are static picture-capture metafiles.

E40: Page 14, 4.4.7 rephrase, since no CGM categories may be other than static picture-capture metafiles. See also item 7.

E41: Page 14, 4.4.8 same as 9 above. See also item 7.

E42: Page 15, sub-clause 4.5.2, 3rd paragraph does not adequately explain the difference between "LOCUS" and "SHAPE".

E43: Page 15, sub-clause 4.5.5, paragraph 7 does not adequately explain how "LOCUS THEN SHAPE" may produce any difference from "SHAPE" alone.

E44: Page 15 sub-clause 4.6 There are several lists in the sub-clause. To which one(s) should the element be added?

E45: Page 20, 4.6.8.1. The phrase "A closed figure is opened..." is worded too much like segments. Use "started" rather than "opened". Likewise, use "finished" rather than "closed" for END FIGURE.

E46: Page 20, 4.6.8.3 - State explicitly whether the sequence "New Region; End Figure" is valid.

E47: Page 40, 4.12.2.1 Provide a reference to 4.12.5 for behavior of COPY SEGMENT.

E48: Page 50 subclause 5.3.11 - The existing shorthand names do not include hyphens, even for multiple word names. Should the addendum have them?

E49: Page 58, sub-clause 5.4.6 - eliminate the "double negative" for clarity.

E50: Section 4.12.2.2. Here and elsewhere, references are made to CGM states not included in Table 3.1. In particular, this sections mentions state GSD which is not in Table 3.1.

ES1: Section 4.12.2.2 The last sentence in this section implies that only the stated elements are allowed in the segment. This is clearly not the case.

ES2: Section 4.12.5 In the example, the multiple attribute changes described by the right column for a single COPY SEGMENT (2) instance should be more explicitly mapped to actions which are taking place. More explanation is needed to clearly illustrate which actions in the segment being copied actually cause the change of state.

U.S. Comments on ISO 8632-2/DAD.1

The U.S. disapproves ISO 8632-2/DAD.1 with the following technical comment:

The technical changes to ISO 8632-1/DAD.1 must be reflected in this part .

Editorial Comments:

None.

U.S. Comments on ISO 8632-3/DAD.1

The U.S. disapproves ISO 8632-3/DAD.1 with the following technical comment:

The technical changes to ISO 8632-1/DAD.1 must be reflected in this part .

In addition, we note the following inconsistency with the with the proposed binary encoding of CGI and request that this inconsistency be addressed jointly by the CGM and CGI RGs of WG3:

CGM and CGI are inconsistent in the specification of precision of integers representing the data types SN, PN, and ASN. The CGI specification uses fixed sized 16-bit integers, which limits each of SN, PN and ASN to 64K unique identifiers. CGM uses integers subject to integer precision.

Editorial Comment:

Clause 7.10, COPY SEGMENT. The enumerated values do not follow the null-value rule that is used in CGM. They should be:

0:	no
1:	yes.

Page 17, new item h). Why is metric scale factor allowed to use fixed format real when scaling mode is not? The text should highlight this difference.

U.S. Comments on ISO 8632-4/DAD.1

The U.S. disapproves ISO 8632-4/DAD.1 with the following technical comment:

The technical changes to ISO 8632-1/DAD.1 must be reflected in this part .

Editorial Comments:

None.

Attachment 2:

US Contribution to the Improved Graphical Text Model

ISO
International Standardization Organization
Organization Internationale de Normalisation

ISO/IEC JTC1/SC24

Computer Graphics

Secretariat: FR of Germany (DIN)

Title: US Contribution to the Improved Graphical Text Model Study

Source (Country, Organization, Committee): ANSI

- Status: ☐ SC24 Output Document
☒ Member Body position
☐ Expert opinion (ALL expert contributions submitted to SC meeting must be authorized by the Member Body).
☐ Working Group output
- Type: ☐ A1: "Immediate" Output Document from WG/SC24 Meetings (to arrive at SC24 Secretariat 2 weeks after meeting).
☐ A2: "Delayed" Output Document from WG/SC24 Meetings (to arrive at SC24 Secretariat 4 weeks after meeting).
☒ B: National/Expert Contributions to WG/SC24 Meetings (to arrive at the WG or SC Secretariat 8 weeks before the beginning of the the WG or SC meeting)
☐ C: Recent Liaison Documents (to arrive at the expert by the start of the WG or SC meeting)
- Type of Distribution: ☒ Through SC24 Secretariat
☐ Direct to Member Bodies

Date of Submission: 2 December 1988

Keywords: Improved Graphical Text Model, Reference Model

2 December 1988

US Contribution to the Improved Graphical Text Model Study

This is a US contribution to the first meeting of the SC24 Study Group on an Improved Graphical Text Model. This contribution is divided into several parts. These are:

- 1) Interpretations and Clarifications of the Terms of Reference (SC24 N172).
- 2) Goals for the Improved Graphical Text Model.
- 3) Requirements for an Improved Graphical Text Model.
- 4) Supporting material.
- 5) Identified issues.

Several annexes provide input documents that may be difficult to obtain otherwise. These are:

Annex 1. SC18/WG1 N616 User Requirements for TCSS (DSSSL) and TPM (SPDL)

Annex 2. ISO DIS 9541, Parts 1-6, Font and character information interchange, 6 June 1988.

Annex 3. ISO DIS 10036, Procedures for registration of glyph and glyph collection identifiers

Annex 4. Examples of "registered" glyphs and commercial "fonts"

Annex 5. *Xerox Interpress Electronic Printing Standard*, Version 3.0, Xerox Corporation, Stanford, CT, December 1985.

Annex 6. SC18/WG8 N715 Standard Page Description Language, Working Draft 4, December 1988.

We interpret the first paragraph of N172 this way:

Conduct a study to develop an Improved Graphical Text Model that will meet the graphical text requirements of a wide range of applications including, but not limited to:

- Office document creation, printing and exchange;
- The creation, printing and exchange of published documents;
- Technical drawing and illustration creation, printing and exchange;
- Graphics arts and presentation graphics; and
- Presentation entities within product data.

Furthermore, this study should consider the requirements for interworking between implementations of graphics standards and standards in other areas. As far as text is concerned, these other areas include, but are not limited to:

- Office and publishing systems,
- External representation of product definition data, and
- Open systems.

We suggest that the list of documents given in N172 be clarified as follows:

1) In the area of current computer graphics practice, the following document describing the "Hershey Fonts" should be considered:

- Hershey, Alan, *A Contribution to Computer Typesetting Techniques*, NBS Special Publication 424, April 1976.

2) In the area of related SC18 work, the material in Annexes 2, 3, 4 and 6 should be considered, as well as the ODA/ODIF Draft International Standard (DIS 8613), especially Part 6, Character Content Architecture.

3) In the area of available descriptions of commercial systems, the material in Annex 5 on the Xerox Interpress "system integration standard" and the following published (and widely available) documents should be considered:

- Adobe Systems Incorporated, *PostScript Language Reference Manual*, Addison-Wesley Publishing Co, Reading, MA, July 1985.
- Adobe Systems Incorporated, *PostScript Language Reference Tutorial and Cookbook*, Addison-Wesley Publishing Company, Reading MA, December 1985.
- Harrington, Steven J. and Robert R. Buckley, *Interpress; The Source Book*, Brady, New York, NY, 1988.
- Knuth, Donald E., *Computers & Typesetting*, Volumes A-E, Addison Wesley, Reading MA, 1986

- Karow, Peter, *Digital Formats for Typefaces*, URW Verlag, Germany, 1987.

The SC18 user requirements documents in SC24/WG1 N7 are out of date. The updated versions in Annex 1 should be substituted.

The last paragraph of N172 discusses schedules. The US notes that only one meeting of this study group is listed in the resolutions of the last SC24 plenary, while the terms of reference calls for 3-4 meetings. To accomplish the work assigned to this study group the US believes that a total of 3 meetings is needed. The additional two meetings might be scheduled as follows:

- a meeting in conjunction with the April 1989 meetings of the Product Data Geometry and CGM extensions study groups; or
- a meeting in late July 1989 in conjunction with Reference Model Rapporteur group or early August 1989 in conjunction with the New API study group.

The US interprets the dates given in N172 for output document availability as requiring that the output documents produced by the study group be circulated to SC24 for review prior to the October 1989 SC24 meetings.

2. Goals for the Improved Graphical Text Model.

1) The model should support the identification and specification of important attributes of fonts, characters and text for the purposes of:

- a) font selection and substitution, and
- b) text rendering accuracy,

as further described in Clause 4.

2) Fallback guidelines for font selection should be possible.

3) The model should accommodate all information in DIS 9541. Individual standards based on the model may adopt only appropriate parts of DIS 9541.

4) The model should have as much compatibility as possible with the existing text model used in current computer graphics standards without compromising DIS 9541 compatibility.

5) The model should distinguish between different types of attributes. At least three categories appear useful:

- a) font attributes,
- b) character display attributes, and
- c) text string attributes.

6) Clients of the new model must have a way to determine the extent of text objects at the time that such objects are defined.

7) The model must be available for the next generation of standards.

- 8) The model must be available for use in CGM Addendum 3.
- 9) A description of the model at an appropriate level of abstraction should be merged with the SC24 Reference Model work.
- 10) The model should support the development of standards for all uses of computer graphics. Such uses include graphics arts, publishing, and pre-press systems, as well as traditional business graphics, CAD/CAM, and other scientific, technical, and mathematical applications.
- 11) There should be a single unified model for our entire family of SC24 standards. The model should also support the needs of all application areas that incorporate computer graphics. Graphical text within individual standards should be based on, but need not include all of, the model.

3. Requirements for an Improved Graphical Text Model.

Future API and metafile standards have the following requirements which the Improved Text Model must support:

- 1) It should be possible to have text objects whose text extent is workstation independent. (See subclause 4.2 for additional details.)
- 2) It should be possible to determine the extent of a text object at the time it is defined.
- 3) Attribute changes within text objects should be allowed. For example, it should be possible to underline part of a text string. (See clause 4.1 for additional details.)
- 4) The model should allow exact font selection by standard (registered) font names.
- 5) The model should support additional attributes and characteristics, including:
 - a) scoring (e.g., underline, overstrike);
 - b) kerning control;
 - c) weight (e.g., bold, medium, light);
 - d) posture (e.g., italic);
 - e) subscripting/superscripting;
 - f) typeface design classification (e.g., serif, sans-serif, Latin);
 - g) font family (e.g., Times, Garamond, Helvetica); and
 - h) others, the need for which may be determined in the future.
- 6) The model should allow the construction of complex compound objects, such as those required for mathematical equations.
- 7) The model should allow shielding/clipping to text images
- 8) The model should support 3D text and fonts.
- 9) The model should support multi-line text (for example, by defining the interaction of control characters with the text model.)
- 10) The model should allow explicit control over width as well as height of text. (See clause 4 for additional details.)

11) The model should allow standards to provide successive levels of complexity in their text models. These are needed to enable simple things to be done easily while giving advanced applications access to more powerful features. The current single-level model is too complex for some and too simple for others.

12) The model should allow the layout of text along arbitrary paths.

13) The model should allow automatic font substitution.

14) The model should allow application-definable glyphs.

15) The model should allow access to font metric information:

- a) average or global metrics, and
- b) metrics for each character.

16) The model should allow the specification and application of user-defined transformations at various points in the transformation of text and characters.

4. Supporting material.

4.1 Attribute changes within text objects

It should be possible to construct a single text object that consists of parts with different attributes. In existing API standards this can only be accomplished by interspersing different output primitives, such as Text and Append Text, with attribute change elements. This makes it difficult to edit "compound text objects" and to identify and control the impacts of changes to edited structure elements. It may be appropriate to provide this functionality within the context of a more comprehensive object definition facility. (See subclause 4.5 for further information.)

4.2 Workstation independent extent for text primitives

The association of font indices to fonts and the realization of fonts are workstation dependent. Unfortunately, the extent of text primitives must be known for some operations performed above the Workstation Stage of the Reference Model. One example is the PHIGS modelling clip which cannot be properly performed on text primitives today since their extents are not available at this stage of processing. In metafile standards, blind interchange of quality text requires that generators be able to determine text extent and rely upon it being interpreter independent.

4.3 Specifying the relative importance of attributes

Some applications attach more importance to some text attributes than to others. There are at least two reasons for emphasizing some attributes over others:

- indicating font selection and substitution criteria, and
- specifying control over the accuracy of the rendering.

Current standards provide some support for the second of these goals through use of the *TEXT_PRECISION* attribute. However, no support is provided for font attribute specification which would assist in font substitution or font selection by off-line or downstream text manipulation and/or generation services which may need to emulate the requested textual effect with the most closely matching available facilities.

The following proposals for accomplishing these goals are provided to initiate discussion:

4.3.1 Font Attributes

A function should be provided which associates font names and attributes with the font indices used during font selection. This is analogous to specifying colours with colour selection indices. The function could also be used to download fonts or otherwise make them accessible. One way this facility could operate would be to specify a font name in a *font table* as the CGM now does. The font name (Times Roman, New Century Schoolbook, etc.) itself implicitly define a set of font attributes which could be used as substitution criteria if the requested font name is not available. Such a font table would be workstation and device independent since it only depends on information about fonts whose characteristics are independently known.

Automatic font substitution has implications for both font resources and the font selection process:

- adequate descriptive information in the font resource; and
- mechanisms to allow applications to specify allowable variations on a font request or to specify requests with varying degrees of precision.

For example an application program that is attempting to do automatic font substitution must have access to enough information from font resources to enable it to determine the characteristics of available fonts and compute appropriate "best-approximations." Such approximation algorithms will vary from application to application.

Furthermore, application programs and metafiles need ways to explain the user's intent and desires where font substitution might be performed. For example, a user may desire only a specific named font (e.g., ITC Bookman), may be willing to settle for "similar" fonts when the requested one is not available (e.g., use Adobe System's version of Allied Corporation's Palatino if available; if not, substitute Adobe's version of ITC Times-Roman; if that isn't available, then use any modern serif font; if there are no serif fonts available, then use any modern font; otherwise...) The depth of the substitution list should not be restricted by the model.

4.3.2 Rendering Accuracy

The current *Text Precision* attribute was introduced so an application could provide guidance to the graphics system on the trade-off between accurate rendering of text and efficient generation of text. It is appropriate to provide such guidance to the system. However, the current attribute is inadequate for modern graphics systems since it does not allow the application to indicate the relative importance of various text attributes in achieving its desired effect.

One way of providing additional control over rendering accuracy might be to introduce a technique analogous to that used by the PHIGS Element Search function. Thus, a special type of name set could define an association between text attributes and names. An application could then provide one such name sets which has members whose associated text attributes are to be accurately rendered. Members not specified indicate text attributes whose values the system is allowed to modify as necessary for efficiency or to insure text fits within the extent of the text object.

4.4 Text-related terminology

4.4.1 Definitions from DIS 9541

Text-related terminology is evolving rapidly. Some traditional names, such as *character*, have been found to be too easily misunderstood and are being replaced by more precise terms, such as *glyph*. Some terminology is motivated by administrative considerations, such as the need to develop a clear separation between the traditional "codes and character sets" work of SC2 and the "font" work of SC18. Many of the definitions in the DIS text of DIS 9541 (Annex 2) were reworked at a Special Working Group meeting held in London in September 1988 to harmonize the treatment of fonts among SC2, SC18, SC21, and SC24. The latest definitions are:

font: A collection of images having the same basic design, e.g. Bookman Italic.

font family: A set of fonts of common design, e.g. Bookman.

glyph collection: An identified set of glyphs.

glyph: An abstract graphical symbol independent of any actual image.

glyph image: The set of information defining the image of a glyph in a particular font resource.

glyph shape: The set of information in a glyph representation used for defining the shape.

glyph metrics: The set of information in a glyph representation used for defining the dimensions and positioning of the glyph shape.

font resource: A collection of glyph representations together with descriptive information and font metrics which are relevant to the collection as a whole.

score: A line drawn through a glyph shape parallel to the baseline [over, under, or through the shape.]

4.4.2 Definitions from ISO 2022

The following definitions are extracted from ISO 2022:

bit combination: An ordered set of bits that represents a character or is used as part of the representation of the character.

character: A member of a set of elements used for the organization, control or representation of data.

coded character set; code: A set of unambiguous rules that establishes a character set and the one-to-one relationship between the characters of the set and their bit combinations.

4.4.3 Concerns

The definitions in subclauses 4.4.1 and 4.4.2 are not well reconciled. One goal of the study group should be to do such a reconciliation.

4.5 Relationship of Text to Other Graphical Primitives

The following model of graphical primitives explains the relationship of text to other graphical objects.

<u>Dimensionality</u>	<u>Category</u>	<u>Attributes</u>	<u>Examples</u>
1	Linear	Polyline	polyline, arc, ellipse, splines, compound lines
2	Area	Interior Edge	filled polygons, filled circles, cell arrays, spline surfaces, compound areas (such as triangle strips and quadrilateral meshes)
3	Volumetric	??	cylinder, sphere, block, CSG, compound volumes
n	High-Order	??	blinking primitives in which time is the 4th dimension
1-n	Composite	per prim.	markers, text, annotation text, symbols

In this context, compound primitives are primitives which are composed of multiple instances from the same category. For example, a compound line could be defined in terms of polylines and arc segments with the linetype pattern being applied continuously along the entire compound primitive. Similarly, compound areas are enclosed areas whose boundaries are defined by instances of linear primitives. This would provide, for example, easy definition of boxes with rounded corners.

Composite primitives are primitives comprised of examples from any of the categories. For example, the shape information of text glyphs could be defined in terms of line, enclosed area, or compound volume information depending on the needs of the particular font.

Many typefaces today are defined as compound areas whose interiors are then filled. Stroke fonts are defined in terms of polylines. Similarly, bit-map fonts are defined in terms of cell arrays.

It should be noted that the current *Cell_Array* primitive is constrained to have all cells rendered. More powerful capabilities could be provided by introducing cell array attributes which would specify an "auxiliary" colour and a flag for indicating whether "auxiliary colour"-ed cells are to be rendered or the background is to show through. A cell array can also be considered as a compound primitive composed of a grid of cells with each cell being a filled polygon.

Composite primitives can be defined in terms of primitives from any category. Thus, symbols could be defined hierarchically and user-defined or system-defined glyphs could reference other glyphs to produce glyphs for logos or mathematical expressions.

Graphical transformations apply to all primitives uniformly. Lighting and shading information can be applied using standard rendering techniques. Since composite primitives are composed of other primitives they need not be excluded from realistic rendering operations.

Issue: T4

Should font attributes such as italics be part of a font name, a font attribute, or both?

History:

12-02-1988, Raised by U.S.

Keywords:

Improved Graphical Text Model

Alternatives:

- 1) part of a font name.
- 2) a font attribute.
- 3) both.

Arguments:

- a) Pro1: All necessary resources known before interpretation.
- b) Pro1: Consistent with typographic usage.
- c) Pro 2,3: All possible combinations in a font name would soon become unwieldy.
- d) Con 3: May result in ambiguity if different values are specified in the name and the attributes.

Issue T5:

Should text facilities allow access to attribute groups appropriate to their shape defining primitives?

History:

12-02-1988, Raised by U.S.

Keywords:

Improved Graphical Text Model, glyph shape

History:

12-02-88

Alternatives:

1. yes
2. no

Arguments:

- a) Pro 1: The possible special effects in displaying text would be enhanced.
- b) Pro 2: Text usage is inherently different from that of other primitives.

Attachment 3:

Minutes of the Munich CGM Extension Meeting



ISO/IEC JTC1/SC24/WG1 N 36

Date: 1989-02-24

Project: 1.24.5

ISO/IEC
International Organization for Standardization
International Electrotechnical Commission

ISO/IEC JTC1/SC24/WG1
Computer Graphics Architecture
Secretariat: National Computer Graphics Association

Title: Minutes of the Study Period Meeting for
Extensions to CGM Static Picture Capture Capability
(Munich, Jan. 1989)

Source: Meeting Secretary (A. Mumford)

Status:

- ☐ WG1 Output Document
- ☐ Member Body Position
- ☐ Expert Opinion
- ☒ Rapporteur Group Output

Type:

- ☒ A1 Immediate Output Document
- ☐ A2 Delayed Output Document
- ☐ B National / Expert Contribution
- ☐ C Recent Liaison Document

Distribution Type:

- ☒ Through WG1 Secretariat
- ☐ Direct to Members
- ☐ Through SC 24 Secretariat

Date of Submission: 1989-01-27

Keywords: Minutes. CGM

**ISO/IEC JTC1/SC24 Study Period Meeting for:
Extensions to CGM Static Picture Capture Capability
18-20 January 1989 - Munich, F.R. Germany
Minutes of the Meeting**

Liaison Meeting

The meeting began with a joint meeting between the CGM group and the other groups who were meeting during the same week. These groups were the Product Data Geometry study group and the Improved Text Model study group. The minutes of that meeting are appended to these minutes.

Participants

The participants in the CGM meeting were:

Germany: Moeller, Brandenburg (till 19th pm), Zapomuel (part of the meeting), Schuur (till 19th pm)

UK: Mumford, Francis, Thomas (part of the meeting)

USA: Bono (till 19th pm), Laris, Stoll, McConnell (part of the meeting)

Apologies were received from the Rapporteur, Lofton Henderson. Peter Bono chaired the meeting until Thursday pm.

Aims of the Meeting

These were to follow the new SC24 guidelines (SC24/N171) and to produce a draft requirements document and a draft new work item for consideration by WG1. The Requirements Document and the draft New Work Item will go to the Reference Model meeting in Paris the week after this meeting and to WG1 28th Feb in Darmstadt. The new work requirements of SC24 require an SC24 ballot prior to the JTC1 ballot.

Relevant Documents

SC24/N9 - Requirements Study
SC24/N15 - Proposal for a CGM Addendum 3
SC24/N52 - An Initial Draft of Addendum 3 produced by ANSI
TC188/SC4/N284 - STEP
SC18/WG8/N715 (Rev) - SPDL
SC24/N177 - SC24 Proposed Reference Model

There were no official inputs to the meeting.

The Way Forward

It is hard to define the precise nature of the work as it is so closely tied in with the other study groups. They need to report before the final requirements can be drawn up. There was concern that this may delay the work which is needed in the market place. A revision of the current Addendum 3 will not be circulated with the NWI as it may confuse and pre-empt recommendations of the other study groups. This would not prevent work on the document being carried out by a national body taking account of these discussions in Munich. The document when it was eventually produced taking account of all study group reports would benefit by review at this early stage.

It was agreed that the purpose of the meeting was not primarily to develop the Addendum 3 work based on previous drafts. The aims were far wider and the purpose to recognise requirements rather than to define precisely how these might be met.

Timescales

When drawing up the timescales account was taken of the fact that some countries had difficulty in participating in so many meetings. It was agreed that a tight schedule which took account of other meetings and held them at the same time was preferable to holding separate meetings which required the same experts to attend. Particular attention was paid to the CGM/GKS addenda editing meeting in June/July in Hawaii and the SC24 meeting in October in Brazil. The following timetable is proposed:

28 Feb 89 SC24/WG1 approve output from Munich meeting
3 March SC24 approve NWI Ballot
15 March NWI ballot starts
15 June NWI Ballot closes
3 July Meeting in Hawaii to revise NWI - faxed to SC24
5 July SC24 forwards ballot to JTC1
15 July JTC1 ballot starts
15 Oct JTC1 Ballot ends
16-29 Oct Working Draft prepared at SC24 meeting
April 90 DP/PDAD text prepared
Sept 90 DIS/DAD text prepared
April 91 IS text prepared

This is a very tight schedule. It was considered that it was necessary at least for the first few stages to ensure improved participation.

Liaison is needed between SC24 and SC18 at their meeting in Munich and at the STEP meeting in San Antonio. We need an SC24 rep not necessarily a member body representative.

Action for the Rapporteur to request liaison at these meetings and to ensure participation.

Discussion on the Requirements

The Tucson meeting discussed the relationship with SC18. The recommendations (SC24/N186) include the need for SPDL to translate a CGM into SPDL in a standard way.

ODA allows this and includes CGM in its specification. Areas of overlap need to be solved in a common way.

There was agreement that extensions to the CGM might include publishing, engineering drawing, business graphics as included in doc. N9. Cartographic requirements are also needed. When do we know when to stop adding functionality? The requirements document must state this.

It was agreed that Addendum 3 (or whatever it becomes) should not be 3D and that it should be built on CGM plus Addendum 1. Then possibly extend to 3D for added capabilities.

It was agreed that a closed list of elements might be better for getting a standard produced than less clearly defined requirements. The precise list should be agreed when the study periods come to an end (October 89). It is hard though with the work going on in parallel within SC24 and outside in other ISO groups. The STEP work is in early stages (DP) of standardisation and their timescale is less aggressive.

The discussions as to the precise requirements were based on a consideration of N15. 6 areas of extension were recognised:

1. Advanced 2D requirements

curves - note STEP line extension might be of interest

fine control over line appearance

composite line primitive

user defined line types, hatch styles, markers - also symbols required with the same definition techniques as markers - and glyphs

additional standardized hatch styles

arbitrary text path

are filling methods sufficient? there is a need to make up styles from the other primitives - note CGI Bitmap fill too but this is not device independent

general linear transformation - need for 3*3 matrices

2. Text and Font Model

take the requirements from the study group on text

3. Arbitrary Boundaries

There was concern as to the number of start-end boundary sequences there are in the doc N52 (plus closed figures in Add1 and CGI). It would be better to have a general path element which would then have its action applied to it.

4. Colour Models

This is definitely a requirement and close study is needed of the ODA colour Addendum. Colour interpolation is also needed.

5. Image

SC2/WG8 have work in the area of compression techniques. Their documents are at DIS level and should be used. Can cell array be improved by making it more compact in its encoding

6. Symbols

There is a need for defining symbols and also for external referencing - this is a general requirement for other areas e.g. fonts.

Another requirement might be for directories of pictures to be stored.

(Alignment (N15) left out as nobody could explain it)

There was some concern that a third addendum to CGM might not be the best way to progress the document. Addenda are confusing. This would also be difficult if there is another colour model as RGB is described in many places. A revision would be better but this would have many implications e.g. doing 3D fully.

It was agreed that the extensions work was to address storage capabilities at the same level of the reference model as CGM and Add 1.

STEP/CGM Reference Models

The group discussed the diagrams presented in N257. The discussion was led by Mr Zapomuel. Presentation entities are on the border of the CAD system and the graphics package.

Some comments were made on the specifics of the diagram. CGM and CGI are not necessarily at the right place relative to the new CGI model. the graphics package can also be wide or narrow depending on the implementation (though conceptually some of the layers may still exist). It would be useful to add the PHIGS archive and GKSM in. It was agreed that this was one example of how things fitted together rather than being a definitive statement.

The bell curves on the second diagram were felt to be a useful representation of the position of the various standards. McConnell to take this diagram to the Reference Model meeting. STEP should end where CGM starts. STEP and CGM should replace IGES giving no overlap between the standards. There needs to be a definition of the differences between a drawing and a document.

Reference Model

The Reference Model draft (N177) was presented by John McConnell. The group discussed the model in relation to the CGM and GKSM and the comments are to be fed back to the Reference Model group.

The CGM is a single workstation with the coordinates stored in virtual device coordinates. The Workstation level is thus the most likely point for the CGM to lie. Elements such as pixel array are probably lower and this is likely to be a point of concern in some of the comments on the Add 1 ballot. If we are to say that the CGM extensions are at the same level as CGM can raster ops be added (as currently proposed).

There was some concern that the model could have multiple coordinate systems at the same level. This makes it harder to place CGM. Could CGM be a list of elements which range across a number of levels of the metafile with the Metafile Category being used to identify a particular sort of metafile and thus where it lay in the model. This is not in line with the idea that CGM is well defined in the model.

It was noted that the CGM extensions work is a part of the first generation of standards and thus this model does not necessarily apply. Also there will not necessarily be standards for all levels.

Discussion on the Initial Draft (N52)

The document N52 which was produced by ANSI some time ago was discussed with the experts making points which can be fed into the next draft. The comments are appended to the minutes.

Output from the Meeting

1 These minutes

Action: Mumford to draft and send to Bono for circulation

2 Draft Requirements Document

Action: Group to draft, McConnell to take to Ref Model meetings and then to arrange circulation in WG1 via Bono

3 Draft New Work Item

Action: Group to draft, McConnell to take to Ref Model meetings and then to arrange circulation in WG1 via Bono

4 Letters from the Rapporteur to Reference Model and Requirements groups in WG1 and to WG1 convenor requesting action on the documents.

Action: Bono to draft and send to Henderson

Appendix 1

Comments on The Initial Draft for Addendum 3 (SC24/N52)

This was a very brief discussion but the following points emerged:

1. Advanced 2D Requirements

a) curves

STEP Geometry has the following 2D primitives (Presentation also needs to be checked)

conic: circle, ellipse, hyperbola, parabola

bounded curve: polyline, B-spline, trimmed curve, composite curve

SPDL must be checked to see if the representation of the curves defined in N52 is the most efficient for SPDL.

b) line appearance

STEP parameterises the end point of the line which can be user definable (4.12.5 in STEP presentation)

STEP also has rounded asymmetric line join. These considerations also apply to edges.

c) composite line primitive

There was a general agreement that the same method should be used for shielding and clipping. Too many begin-end pairs appear in N52. Should this also be adopted for closed figure? There are difference here though. Need to look as SPDL paths. There may be conflict with CGM and Add1 though.

d) user defined line styles etc.

There was a feeling that the needs of cartography had not been addressed and that they should be.

e) additional hatch styles

There is a need for styles to be defined from the other primitives. We need a wider capability for filling of areas than just cross hatching. How should this be addressed - what is a hatch?

f) arbitrary text path

the begin-end path comments addressed above apply here too.

g) filling

PostScript has the non-zero winding rule as well as the odd/even filling. (pg 71 red PostScript book). Interpolated filling (going from one colour to another) also seems important.

h) **general linear transformations**

These are not in N52 but would be useful.

2. **Improved Text and Font**

Strong feeling that the work of SC18 and the work of the study group on the improved text model must be the main driving force for the elements in this section.

3. **Clipping and Shielding**

What does text shielding apply to - the character box or the shape? This needs to be addressed in relation to the discussions on glyph definition in the improved text study group.

4. **Colour Models**

There is nothing in N52. This needs to be addressed but it will be one of the hardest things to add in if the text is to be an addendum. RGB is mentioned very many times in the CGM text. The ODA colour model and PHIGS should be used as base documents for the elements.

5. **Imaging**

The reference model makes this a problem. How can the CGM+Add3 be at the same place in the reference model if these more device dependent elements are added in? It was recommended that the work on picture coding standards in SC2 should be the basis for any definition. Do we need Pel Array Clip Rectangle? Does general clipping apply to raster? Arbitrary clipping of raster might be useful too.

6. **Symbols**

Has Add 1 dealt with this? If not, why not?

Attachment 4:

Addendum 3 New Work Item Proposal

Addendum 3 Requirements Document

CGM ADDENDUM 3 NEW WORK ITEM PROPOSAL

Scope

This work comprises a set of elements which will extend the capabilities of the CGM (IS 8632) and CGM Addendum 1 (ISO 8632/DAD.1) to meet additional user requirements.

The following list of capabilities will be addressed by this work:

1) Advanced 2D graphics, to include:

- curves
- fine control of line appearance
- composite line primitives
- user defined line types, hatch styles, and marker types
- additional standardized hatch styles
- arbitrary text path
- filling mechanisms
- general linear transformations

2) Improved text and font support

3) Arbitrary boundaries for clipping and shielding

4) Additional color models beyond RGB

5) Additional raster graphics (scanned image) capabilities

6) Symbols: external reference to "standard" libraries of named symbols

The precise list of elements to be included in this group will take account of the work of the SC24 study groups in the following areas: Improved Graphical Text Model, Product Data Geometry, new API for graphics, PHIGS BR, Reference Model for Computer Graphics, and also the GKS Maintenance work.

Purpose and Justification

The purpose of this work is to extend the CGM and CGM Addendum 1 to fulfill additional 2D picture storage and retrieval requirements. CGM users have found that in some application areas the present standard provides a general framework that is suitable but lacks some functionality required by these applications. These areas include engineering drawing, the preparation of graphic arts quality presentation materials, cartography, and technical publishing.

SC24 has recognized the need to serve these application areas and to meet the requirements which go beyond those currently specified in the standards developed, and being developed, within SC24. The precise requirements will be the result of the deliberations of the study groups set up by SC24 to consider an Improved Graphical Text Model, Product Data Geometry, and a new Application Programming Interface for Graphics. It is considered that meeting these requirements is essential if the CGM is to continue to be used in the areas recognized above.

It is essential that this work uses some of the solutions adopted within related ISO Standards activities, for example font definition standards, colour models, and product data exchange standards.

Co-operation and Liaison

Work will progress within ISO/IEC JTC1/SC24 in liaison with:

ISO/IEC JTC1/SC18
ISO TC184/SC4
ISO/IEC JTC1/SC2

Relevant Documents to be Considered

CGM, IS 8632 (Parts 1-4)
CGM Addendum 1, IS 8632/DAD-1 (Parts 1-4)
Font and Character Information Exchange, ISO DIS 9541 (Parts 1-6)
ODA, ISO DIS 8613 Colour Addendum
PHIGS BR
Reference Model for Computer Graphics
STEP, ISO TC184/SC4/N284

Program of Work

The following schedule is proposed for this work:

Oct. 1989	- NWI approved, WD prepared
April 1990	- DP or PDAD text prepared
Sept. 1990	- DIS or DAD text prepared
April 1991	- final text prepared

Assignment of Work

SC24 requires that this work item, if approved, be assigned to SC24/WG3.

Requirements Document for Extensions to The Computer Graphics Metafile

1.0 Introduction

It has been recognized that the current CGM Standard needs to be extended in order to effectively fulfill the picture storage and transfer requirements of engineering drawings, graphics arts and technical publishing. The purpose of this document is to identify the requirements for extension explicitly, as specified by the procedures stipulated in ISO/IEC JTC1 SC24 N171.

2.0 Applications

2.1 Application Areas With Further Requirements:

a) Engineering Drawings

The CGM is currently being used in engineering applications to store and exchange picture information. This area of use has found the CGM lacking in functionality which, if present, would increase the efficiency of storage. Examples of required elements include curve definitions, such as B-spline and conics, and improved text capabilities. In other areas more control is required. Examples of this include control of line appearance such as line ends. There are also requirements for specifying and referencing symbols.

b) Graphics Arts

For graphic arts quality a higher degree of control over the rendering of graphical and text objects is required. Line attributes are required to control the rendering of line endings and line joins and user-defined line types are also required. There is a requirement to be able to print text along an arbitrary path and to specify fonts and text attributes in a more precise manner. Further colour models and raster encodings are also required.

c) Technical Publishing and Illustrations

One of the main intentions of the CGM Standard is to support the requirements of this application area. Though the CGM has been used successfully in the technical publishing and illustration environment, it is felt that some of the inherent limitations in the CGM Standard have curtailed its acceptance in the marketplace. Several enhancements to the CGM have been identified to address these limitations, including but not limited to: line appearance control, arbitrary text path, enhanced text capabilities, and additional colour models.

2.2 Studies of Application Requirements

a) Product Data Geometry Study Group Report - PDG SG

The purpose of this study group is to look at the relationships between Product Data Geometry (PDG) and Computer Graphics; its membership should be drawn from experts from the Computer Graphics Standards and Product Data Geometry Standards communities. In discussing these relationships at the first meeting in Munich, it was found that there exists some overlap between the stated goals of the PDG Standards and the CGM. The SG concluded that in certain areas the CGM, if extended, could be utilized to fulfill those goals. Some of the requirements for extension included complex curves, additional user-definable attributes, additional font/text functionality, and symbol libraries.

b) Text Study Group Report - Text SG

Given that SC24 has recognized that the current text model needs review, this study group has been chartered to look at what enhancements are necessary and/or desired for the existing graphical text models. The main conclusion of this SG at the Munich meeting was that the Font Standard work taking place in SC18/WG8 should be the primary source of technical input for improving the graphical text model. The SG also uncovered some additional requirements for text in studying the

STEP standard. Some of the enhancements identified for the CGM included font definition and referencing, font substitution, and rendering.

c) CGM in the Real World Workshop - CRWW

This Workshop, involving implementors of the CGM, met in September of 1987 to discuss the successes and difficulties encountered when implementing the CGM Standard. At this workshop, several common problems and shortcomings of the CGM were identified. Some of these included, but were not limited to: the inadequacy of the text model, the lack of advanced 2-D graphics primitives, and the need for additional colour models.

d) User Requirements Study of Publishing and Technical Drawings - WG1 N9

This report was produced under the auspices of the U.S. DoD Computer- Aided Acquisition Logistics Support (CALS) Office. CALS is a major initiative intended to automate the production, exchange and publishing of product support information. Product support information in CALS includes product data, raster graphics, 2-D technical drawings and text. This study (WG1 N9) specifically addresses the extension requirements for the CGM to support the technical illustration exchange and storage needs for CALS. Several needed enhancements are identified, including bezier curves, line cap and join, enhanced text, shielding, interpolated fill, and additional colour models.

e) Input into the GKS Review - Cartographic Requirements (CR)

This document is a requirements statement put together by experts from the cartographic industry regarding the enhancements needed to GKS to support cartographic applications. These enhancements are also required for the CGM, and include: geometrically specified patterns, arbitrary clipping regions, and external symbol libraries.

3.0 Required Capabilities

The following table provides a list of the features that were identified as requirements in the studies mentioned in Section 2. The features are listed along with the study from which they originated:

—FEATURE—	—REQUIREMENT SOURCE—
Advanced 2-D graphics:	
- curves	PDG SG, CRWW, WG1 N9, Text SG
- line appearance control	PDG SG, CRWW, WG1 N9
- composite line primitives	WG1 N9, CRWW
- user-defined line, hatch, marker types	WG1 N9, PDG SG, CRWW
- additional standardized hatch styles	WG1 N9, PDG SG
- arbitrary text path	PDG SG, Text SG, WG1 N9
- geometrically specified patterns	PDG SG, CR
- general transformations	PDG SG
Enhanced text and font capabilities:	
- font/glyph definition	Text SG, PDG SG, CRWW WG1 N9
- font referencing and substitution	see above
- rendering	see above
- additional text attributes	see above
Arbitrary boundaries:	
- clipping	WG1 N9, CR, CRWW
- shielding	PDG SG, WG1 N9
Colour models:	
- CIE, CMYB, named colours, etc.	WG1 N9, CRWW
- interpolated fill	WG1 N9

Additional raster graphics:
- coding/compression techniques

WG1 N9

Symbols:

- external lib.
- use. defined internal lib.

WG1 N9, PDG SG, CR
see above

4.0 Examples

4.1 Advanced 2D Graphics

The CGM in the Real World Workshop examined the issue of advanced 2D graphics and concluded that the CGM lacked capabilities to effectively meet some advanced user needs. For example, for engineering drawings it is difficult, if not impossible, to effectively represent some higher-level constructs in the CGM, such as splines and curves. Though such constructs can be simulated with simpler primitives in the CGM, it is frequently impossible to maintain accuracy and visual continuity and still retain device independence. A list of additional functional requirements for advanced 2D graphics follows, with examples of the realization requirements:

a) Curves

Curves include the general class of curved line elements that are more complex than the existing circular and elliptical arc elements, such as:

- Bezier curves
- Rational B-splines
- Parametric spline curves
- Conics, and conic arcs

b) Fine Control of Line Appearance

This includes the additional line attributes of cap, mitre, and join.

c) Composite Line Primitive

This consists of a line composed of both straight and curved line segments.

d) User Defined Line Types, Hatch Styles, and Marker Types

An example would be the ability to define a line type, such as the stitched, center, or phantom line types frequently used by engineering drawing applications.

e) Arbitrary Text Path

Text drawn along a composite line path.

f) Filling Mechanisms

Interpolated Fill, which is a fill comprised of colours interpolated linearly between two reference colours.

g) General Linear Transformations

3 x 3 (and 2 x 3) transformation matrices to allow for affine and projective transformations

4.2 Enhanced Text and Font Capabilities

The enhanced text and font capabilities should accommodate most of the information in the ISO DIS 9541 Font and Character Information Interchange Standard. This standard defines a font resource architecture to support text generation, interchange and presentation. A font resource has to provide sufficient information to characterize and identify a font in order to allow font referencing and font

substitution. In the latter case, mechanisms to specify allowable variations on a font request or requests with varying degrees of precision should be provided.

Font and glyph definition requires attributes in addition to those used in the current computer graphics text model, such as typeface design specification, posture, and kerning control. However, to support features such as arbitrary text path, information currently not included in ISO DIS 9541 would have to be provided.

Techniques used in modern graphics systems must be provided to guide the interpreter of a metafile as to the desired rendering accuracy to indicate the relative importance of various text attributes for achieving a desired effect.

5.0 Constraints

The work which is proposed to meet these requirements will be based on the CGM standard and CGM Addendum 1. It is intended that this extension will occupy the same point in the SC24 Reference Model as the original standard. This means that the extension will produce a metafile suitable for the storage and retrieval of 2-D picture information, and will not address 3-D or dynamic capabilities. It is the goal of this work that it will be useable by related standards including the font standardization effort, ODA, and STEP.

Close working with the GKS Maintenance group is essential. Where this group defines functionality which is the same as that proposed for this extended metafile work, then the groups should work together to produce functionally identical specifications and encodings. The progress of this work will be measured by its relationship to these other standards and their resulting compatibility.

The New Work Item proposal defines the areas of extension and notes the need to adopt the results of the study groups of SC24. It is strongly recommended that these functional requirements are turned into realization requirements as defined by SC24/N171 on acceptance of the NWI and as the working draft is being prepared. This means that if the proposed schedule is adhered to, then the general requirements will be turned into a list of elements to be included in the new work at the SC24 meeting in October 1989. This does limit this work to extend the CGM to include only those requirements which have been accepted at that time. This should ensure that progress can be monitored, problem areas identified early in the project, and these urgent needs for CGM users met in a timely fashion.

6.0 Bibliography

STEP - TC184/SC4/N284
IGES V3.0
EDIF
PHIGS BR - JTC1/SC24/N224
GKS Review - Cartographic Requirements
Font Standard - ISO/DIS 9541
ODA - ISO 8613
SPDL - JTC1/SC18/WG8/N715
Liaison Statement to SC18 Regarding SPDL - SC24/N208
Terms of Reference for Improved Text Model - SC24/N172
U.S. Contribution on the Improved Graphical Text Model
GKS - ISO 7942
PostScript - Reference Manual and Tutorial
Interpress - Introduction
Modern Drafting Practices and Standards Manual
CALs - MIL-STD-1840
Line Conventions and Lettering - ANSI Y14.2M 1979
Dimensioning and Tolerancing - ANSI Y14.5
CGM in the Real World - Springer Verlag, 1988

Attachment 5:
Final Report of CGM Extension Study Period

Final Report of CGM Extension Study Period
Lofton Henderson, Rapporteur
10 August 1989

1. INTRODUCTION

At its July 1988 meeting, ISO/IEC JTC1/SC24 passed a number of resolutions establishing study groups and study periods on future computer graphics standards work within SC24. Resolution 9 established a study period to examine the need for further extensions to the Computer Graphics Metafile standard (CGM, ISO 8632/1987). Further extensions had been proposed in order to meet the requirements of application areas such as engineering drawing, technical publishing, and graphic arts, and had commonly been referred to as "Addendum 3."

One set of extensions, Addendum 1, was already in an advanced state of processing. A second set, Addendum 2 (for 3D), was technically an active project but was without a document editor and was not progressing. Both of these projects were being processed under ISO rules for addenda, according to resolutions at the SC24 plenary in September 1986. These resolutions established the addenda projects without New Work Item review and ballot.

It was determined by SC24 that the proposed Addendum 3 would be progressed according to the new NWI procedures in N171. Under these procedures the need for such a project would be studied by a study period, a requirements document would be generated and reviewed by the requirements rapporteur group within SC24/WG1, the position of the proposed work in SC24 reference models would be reviewed by the reference models group within WG1, and finally an NWI would be generated and balloted.

Key technology areas of Addendum 3 include advanced graphical text capabilities and advanced geometric primitives. Because these technology areas are expected to be shared among several of the next generation of SC24 standards, study groups were established to examine each of the areas. The Addendum 3 study period was directed to coordinate closely with these two groups.

2. SUMMARY OF RESULTS

The Study Period concluded that the extensions referred to as "Addendum 3" are needed and must be expedited if CGM is to continue to be an important standard.

The provisions of N171 were followed. A requirements document and reference model statement were produced and submitted to WG1. A New Work Item proposal was drafted and submitted to SC24 for a three month ballot. The ballot passed with no negatives, and a slightly revised NWI is currently being balloted in JTC1. A Working Draft of Addendum 3 has been produced and is currently undergoing a three month comment period in SC24.

3. HISTORY

3.1 Munich Meeting

The first meeting of the Study Period was held in Munich, F.R. Germany. 18-20 January 1989. It was held in parallel with the initial meetings of the Text Study Group and the Product Data Geometry Study Group. These two groups met for the first half of that week, and the CGM group met for the latter half. Many of the attendees participated in one or the other of the technology study groups and then the CGM meeting.

The goals of the meeting were to ascertain the need for CGM Addendum 3 and execute the first steps in the new NWI procedures as detailed in SC24/N171 — produce a requirements document, a Reference Models statement, and a draft NWI, and forward these documents to the appropriate groups.

The CGM meeting was attended by:

Germany: Brandenburg, Moeller, Schuur, Zapomeul.

UK: Francis, Mumford, Thomas.

US: Bono, Laris, McConnell, Stoll.

The meeting was chaired by Bono in the absence of the rapporteur Henderson.

There were no official inputs to the meeting, but base documents referenced in the meeting call included some requirements studies and a draft NWI proposal.

The output of the meeting consisted of:

- Minutes (WG1/N36).
- Draft Requirements Document;
- Draft New Work Item proposal;
- Letters to the WG1 Reference Model RG and the Requirements RG requesting action on these documents;

The detailed results of the meeting are contained in these documents. The final versions of the NWI and Requirements Document are included as an attachment to this report. Significant points of the meeting are summarized here.

The dependence of the final output of this Study Period on the outputs of the Text and Product Data groups was recognized. Concern was expressed that this dependence could slow the work down considerably. It was agreed to progress the work as much as possible, but there would be need for looking at the final outputs of the two groups when they become available.

There was some discussion of how to progress the project — should it be an addendum? The general feeling is that it should not, because this would be too unwieldy. Decision on the exact method of processing will be deferred until a later date.

A schedule was derived which would result in IS text in April 1991. This included a tight schedule to complete all of the required NWI processing before the SC24 meeting in October 1989. By the end of that meeting there should be a complete list of requirements and a closed list of elements.

The next and final scheduled meeting was set to be in Waikoloa, Hawaii. 26-28 June. If the schedule were kept this meeting would process the results of the SC24 ballot on the NWI proposal.

3.2 Between Munich and Waikoloa

The Draft NWI and Draft Requirements documents, with cover letters from the rapporteur, were sent to the Reference Model Rapporteur Group and the Requirements Rapporteur Group as per N171.

The Reference Model group responded informally. Because there are no formal liaison documents, that response will be reviewed here. The Reference Models RG pointed out that there could be some problems with asserting that the proposed CGM Addendum 3 occupies the same "level" in the reference model as CGM. This problem was seen during the processing of CGM Addendum 1, which at that time contained a low-level device dependent formulation of the Pixel Array element (it was removed from Add.1 for this reason).

The Reference Model RG felt that the same problem could arise in Add.3, particularly if care is not taken in formulating the additional imaging capabilities. They pointed out that if the realization of the proposed functions of Add.3 straddled stages or levels in the Reference Model, then CGM would have to make a choice: either CGM-plus-addenda is at a well defined stage in the pipeline and any functions to the contrary would be proscribed and removed; or that principle no longer pertains and CGM-plus-addenda would be viewed as a set of encoding techniques to be applied to objects at different stages in the pipeline.

The general consensus of the CGM Study Period and the WG3 CGM Rapporteur Group is that the CGM standard should remain as a static picture capture mechanism whose features can be placed at approximately the current CGM level in the pipeline. This is an issue that CGM must keep in mind while progressing any addenda. Neither CGM nor Reference Models have been able to precisely specify what criteria determine whether functions are at the same level or stage. This question is bound to be subject to some interpretation, and will likely always generate differing opinions. But at least there should be some clustering around a stage in the pipeline, and elements which create recognizable technical problems (such as the CGI PIXEL ARRAY that was originally included in Add.1) should be avoided.

The Requirements Rapporteur Group of WG1 did not generate a formal response to the Draft NWI and Requirements documents either, apparently due to scheduling difficulties. There is in the WG1 document register (WG1 N41) a U.S. position paper. A couple of comments in this paper seem pertinent. In the area of constraints, the CGM requirements document does not adequately express: requirement for the availability of resources; requirement for timeliness (must be done by YYMM). These points must be kept in mind by the Metafile Rapporteur Group.

Because there were no requirements for change from either of the WG1 rapporteur groups, the same two documents that were produced by the CGM Study Period at Munich therefore went to the WG1 plenary meeting in late February. Recommendation 3 of that meeting recommended that the SC24 Secretariat immediately conduct a three month ballot on the NWI so that the results could be processed at the final scheduled meeting of the Study Period, 26-28 June in Waikoloa, Hawaii.

The three month ballot was commenced as planned, and terminated on 15 June.

3.3 Waikoloa Meeting

The Waikoloa meeting was the last scheduled meeting of the CGM Extensions Study Period. Its purposes were to:

- process the results of the NWI ballot;
- revise, if necessary, the NWI and the Requirements Document;
- express these to the SC24 Secretariat, to be forwarded for a 3-month JTC1 ballot to close before Brazil.
- do technical work on the Working Draft.

The meeting was held in parallel with a meeting of ANSI X3H3. It was attended by:

Germany: Eckhard Moeller.

UK: Alan Francis, Anne Mumford.

US: Bruce Garner, Lofton Henderson (Rapporteur), Mike Laris (provisional Document Editor), Lori Pearce, Harold Schechter.

Anne Mumford is succeeding Eckhard Moeller as rapporteur of the WG3 Metafile RG, which will be processing Addendum 3.

There were no formal inputs to the meeting. The U.S. produced and circulated to attendees a baseline document to serve as a starting point for a working draft. The SC24 Secretariat sent NWI ballot results by Fax. The results: 9 approve, 2 approve with comments, 0 disapprove, 0 abstain. Comments were received from:

U.K. — stressing importance of liaison and harmonization with closely related groups;

Japan — making suggestions that the NWI and Requirements Document be improved by providing more details.

The comments of the U.K. were thought to be addressed adequately in the current document. Some time was spent discussing the comments of Japan. The additional detail suggested was deemed to amount to a justification of the requirements that had been generated from various sources. While such more detailed information would in fact be of interest, it was thought to not be a necessary component of the documents required by N171. In any case the group concluded that it did not have

the resources to develop the additional information, as that information would essentially have to come from the source of the requirement. As can be seen from the Requirements Document, those sources are diverse and were not generally present at the meeting.

Consequently, minor improvements to the documents were made and the rapporteur sent these on to the SC24 Secretariat for commencement of the SC24 ballot.

By the time of this meeting there had been no further activities in either the Text Study Group or Product Data Study Group, and so the NWI and Requirements Document still contain references to the work of those groups as the source for specific functional requirements.

There was a liaison meeting between the CGM group and a number of outside experts interested in STEP, PHIGS, PHIGS+, and reference models. The topic was metafile requirements that are derivable from STEP and PHIGS(+). As far as Add.3 is concerned, there appear to be few new requirements, or at least few concrete requirements were generated, because Add.3 is explicitly a 2D metafile. The model for STEP, computer graphics standards, and metafiles that seemed to get the greatest consensus involves a 3D file. PHIGS+ is seen as serving presentation and graphical requirements of STEP, and it was felt that a PHIGS+ workstation state capture file (as opposed to purely graphical capture) was the sort of metafile support needed. Meeting the metafile requirements of STEP would be accomplished through Addendum 2, and would involve some modification of the scope of Add.2.

The group discussed resources. It is agreed that there are not sufficient resources to process two projects (Add.2 and Add.3). The current WG3 Metafile RG members tend to be more interested in the 2D work (Add.3) than 3D work. This was not an official position, but an informal survey. It is clear that both projects cannot be progressed satisfactorily if more people are not available. A solution could involve collaborative work with WG2 for 3D metafiles.

It was generally agreed by the group that Addendum 3 should not be progressed as an addendum, at least not beyond the earliest stages. Rather, it and Add.1 should be folded into the CGM standard to produce a complete new document. This issue will be addressed by the WG3 Metafile RG when it commences work on the project.

There was time to work on the Add.3 baseline document during the meeting. The group divided into technical subgroups examining particular topics. As a result the baseline document was advanced sufficiently that the group feels it is suitable for the status of Working Draft. It was sent to the SC24 Secretariat for immediate circulation, with the hope of getting early comments that could be processed at Brazil.

4. CURRENT STATUS & PENDING MATTERS

The NWI and Requirements Document are before JTC1 for a 3-month ballot. It is hoped that this will close before Brazil. Unfortunately some long delays have been reported in getting previous items through JTC1, and this may adversely affect the Add.3 schedule.

The WG3 Metafile Rapporteur Group will have to ascertain the level of resource commitment for Add.3, presumably at the Brazil meeting.

The NWI and Requirements Document commit Add.3 to a certain level of coordination with the output of the Text Study Group and the Product Data Geometry Study Group. The final reports of these two groups will have to be examined to see if any further specific requirements for Add.3 are implied.

Before the Working Draft stage is complete, the Metafile RG must finalize the agreed set of functions for Add.3 – this should be done at Brazil.

A Working Draft is currently being circulated for NWI comment. Early comment is hoped for by Brazil, so that work may take place at Brazil.

5. COMMENTS ON THE STUDY PERIOD

Herein are personal comments and observations of the rapporteur, both on the new procedures and on this Study Period.

The procedures outlined in N171 are excellent in principle. Some improvement was absolutely needed over the processes by which CGM itself was standardized, and by which Add.1 and Add.2 were undertaken. In the case of CGM perhaps 1-2 years of delay was incurred by fundamental disagreements over what was being standardized. In the case of Add.2, the scope and purpose have shifted several times, the effort is has not been sufficiently driven by agreed requirements, and in consequence little progress has been made. In the case of Add.1, these problems were potentially present again, but fortuitously there was reasonable consensus among those working on the project and good progress was made.

The result of the procedures is that 11 voting nations have endorsed the scope and purpose, and the high level functional definition of Addendum 3, and no voting nations are opposed. The hidden disagreements that have interfered with previous metafile work should not be a problem for Addendum 3.

There have been problems with the process however. The main problem is that it has taxed participants' resources too heavily. There were too many meetings and too much requirement for liaison for the resources available.

As a secondary effect of the resource shortage, certain steps specified in N171 were missed or passed as formalities. No formal input ever came back to the Study Period from the Requirements RG or the Reference Model RG before the Requirements Document and NWI went out to SC24 for ballot. This was due in one case to a liaison that was missed for lack of time and in the other case to a meeting which had to be cancelled.

The content of the Add.3 project was made dependent upon the output of two technology study groups. All three groups started off with a high activity level. One of the technology groups resulted in specific items that could be adopted by the Add.3 NWI; the other did not generate such. In both cases, the work of Add.3 Study Period cannot be complete until the final reports of the two groups are available, at Brazil, 16 months after the process commenced.

The Add.3 Study Period was somewhat fortunate in the timing of events. Once the initial meetings happened in January, the meetings of the Reference Model group, the Requirements group, WG1 plenary, and the SC24 AG happened very soon after. Other projects following the procedures of N171 could be less fortunate and could incur significant delay. If all of the dependencies, those required by N171 and those required by technical liaison, had been rigidly observed it is easy to imagine that the process could take significantly longer than this Study Period.

In the balance the process was beneficial. New standards should be driven by agreed requirements and should have consensus on scope before work begins. However given the current environment and pressure on resources the procedures need to be simplified and streamlined.

Attachment 6:
Addendum 3 Working Draft

ANSI X3H3

Information Processing Systems --

Computer Graphics --

Metatile for the Storage and Transfer
of Picture Description Information

Part 1

Functional Specification

(Clause 3)

Addendum 3

Draft Document 2.0

Sub-clause 3: add or change the following entries:

colour model: A specification of a 3D colour coordinate system and a 3D subspace in the coordinate system within which each displayable colour is represented by a point. Some colour models include a fourth, redundant, dimension to allow the independent representation of black. For the purpose of ISO 8632 colour model refers to one of RGB, CIE $L^*u^*v^*$, or CYMK.

colour selection mode: Indicator as to whether colour selection is to be direct (by specifying a colour value) or indexed (by specifying an index into a table of colour values). See COLOUR VALUE.

colour representation method: Indicator as to which of three colour models (RGB, CIE $L^*u^*v^*$, CYMK) or spot colour is being used to represent colour values. See COLOUR VALUE, SPOT COLOUR.

colour value: The character string (for spot colour) or values of the point components (for colour model) describing a colour.

RGB: An additive colour model, well matched to colour display monitors, whose values are defined by the normalized weights of Red, Green, and Blue components.

CIE $L^*u^*v^*$: A colour model defining an absolute colour space based on colour matching experiments whose components are L^* (Lighness) and u^* , v^* (Chromaticness).

CYMK: A subtractive colour model, common in the printing industry, which has cyan, magenta, yellow, and black components.

Reference Colour Model: Basic colour model within CGM relative to which relationships to specifiable colour models (RGB, CYMK, and CIE $L^*u^*v^*$) are calibrated. The reference colour model is defined by the CIE 1931 standard colorimetric system (XYZ).

spot colour: An exactly defined colour with a registered name.

ANSI X3H3

Information Processing Systems --

Computer Graphics --

Metafile for the Storage and Transfer
of Picture Description Information

Part 1

Functional Specification

(Clause 4)

Addendum 3

Draft Document 2.0

Page 10

Subclause 4.3: Add the following to the list of elements given in the first paragraph of this clause:

COLOUR REPRESENTATION METHOD
FONT DEFINITION
FONT ATTRIBUTES
CHARACTER KERNING MODE
CHARACTER KERNING TABLE

Page 10

Subclause 4.3.2.1: Add the following to the list of elements given in the second paragraph of this clause:

COLOUR REPRESENTATION METHOD
CONIC ARC
CONIC ARC TRANSFORMATION MATRIX
PARAMETRIC SPLINE CURVE
RATIONAL B-SPLINE CURVE
RATIONAL B-SPLINE CURVE CLOSED

Page 11

Subclause 4.3.2.2: Add the following to the list of elements given in the second paragraph of this clause:

COLOUR REPRESENTATION METHOD
FONT DEFINITION
FONT ATTRIBUTES
CHARACTER KERNING MODE
CHARACTER KERNING TABLE

Page 11

Add the following as subclause 4.3.4

4.3.4 Font Elements

The FONTMETRIC DEFINITION element is provided to all permit exact typographic placement of the character glyphs specified within a text string. Using FONTMETRIC DEFINITION, the initial character in a text string would be placed at the specified coordinates, and each subsequent character would be offset by the width and right side bearing of the previous character and by its own left side bearing. If character kerning is also in effect, then the inter character space would also be adjusted by the specified kern value. The character height and offset from the baseline are used to determine interline string placement, ascent, and descent.

The FONT ATTRIBUTES element can be used to select a best fit font if an exact match is not available on a specific device.

Page 12

Subclause 4.4.2, first line:

Change "direct (RGB) colour"

into "direct colour"

Page 14

Subclause 4.4.6, second paragraph, first line:

Change "RGB"

into "a direct colour"

Page 14

Subclause 4.5.2: add the following to the end of the subclause

The IMAGE APERTURE is not affected by the setting of the CLIP INDICATOR element. Aperture setting for pel array elements is assumed to be always on. The default IMAGE APERTURE is listed in clause 6.

Add the following as subclause 4.5.3:

4.5.3 Aperture.

Selection of the region of interest within a pel array, whether clipped by the CLIP RECTANGLE or not, is accomplished using the IMAGE APERTURE. Since the IMAGE APERTURE 'mode' is always assumed to be 'on', the display of all pel array elements is always considered to be controlled by an aperture setting. The default image aperture is listed in Clause 6. The IMAGE APERTURE element thus affects all subsequent pel array elements that follow in the metafile until the aperture is overridden by the appearance of the next IMAGE APERTURE element.

Page 15

Subclause 4.6: add the following to the list:

CONIC ARC
COMPRESSED PEL ARRAY
TILED PEL ARRAY
PEL ARRAY REFERENCE POINT
PARAMETRIC SPLINE CURVE
RATIONAL B-SPLINE CURVE
RATIONAL B-SPLINE CURVE CLOSED

Page 15

Subclause 4.6: add the following to "The line elements are:" list:

CONIC ARC
PARAMETRIC SPLINE CURVE
RATIONAL B-SPLINE CURVE

Page 16

Subclause 4.6: add the following to "The filled-area elements are:" list:

RATIONAL B-SPLINE CURVE CLOSED

Page 16

Subclause 4.6: change "The cell array element is:" to "The cell array elements are:" and add the following to the list:

COMPRESSED PEL ARRAY
TILED PEL ARRAY
PEL ARRAY REFERENCE POINT

Page 16

Subclause 4.6.1.1: change subclause to read the following:

4.6.1.1 Description. There are two general line elements - POLYLINE and DISJOINT POLYLINE - four line elements relating to circles, ellipses and conic arcs, and two elements that relate to spline curves.

Page 16

Subclause 4.6.1.1: change the end of the subclause:

CONIC ARC	generates a parabolic, hyperbolic or elliptical arc; the parameterization of the arc(s) is described in 5.6.X.
XXX SPLINE CURVE	generates a single spline curve; two separate parameterizations of the spline curve are possible; these are described in 5.6.X and 5.6.X+1

Page 16

Subclause 4.6.1.3: change the last sentence of the subclause to read:

"The ARC and SPLINE primitives...

Page 17

Subclause 4.6.4.1: change the second sentence of the subclause to read:

"In addition there are seven elements that..."

Page 18

Subclause 4.6.4.1: add the following to the end of the subclause:

RATIONAL B-SPLINE CURVE CLOSED	generates a closed B-spline curve the set of styles is the same as for POLYGON.
--------------------------------	---

Page 18

Subclause 4.6.4.3 2nd paragraph: change the sentence to read:

"The circular, elliptical and B-spline fill primitives..."

Page 18

Add the following as subclause 4.6.5.1:

4.6.5.1 Pel Array Elements

COMPRESSED PEL ARRAY

represents a rectangular binary image compressed according to the CCITT T4 or T6 facsimile recommendations: two parameterizations are possible, one corresponding to the normal facsimile-size image, and a tiled format for large images; the elements are described in 5.6.X and 5.6.X+1.

TILED PEL ARRAY

represents a series of equally sized, contiguously positioned individual raster images, or "tiles". The first tile is placed at the PEL ARRAY REFERENCE POINT, and then the tiles are placed in sequence in the direction of the pel path and line progression as shown in figure X. The tiles are numbered by the tile index contained in the pel array identifier parameter of each tile.

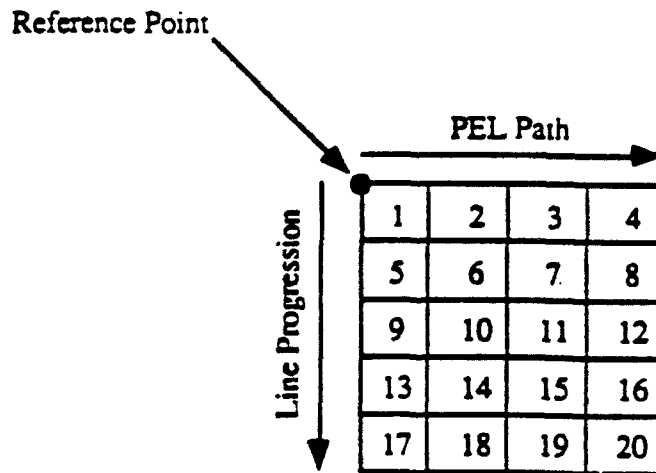


Figure X Ordering and layout of tiles by index

4.6.5.1.1 Attributes. The orientation and dimensions of the pel array elements is controlled by the PEL ARRAY ORIENTATION and PEL ARRAY DIMENSIONS elements. The PEL ARRAY ORIENTATION element specifies the the direction of the layout of pels, i.e. the pel path, in 90 degree increments, relative to the VDC axis. This element also specifies the direction of the layout of lines of pels, i.e. the line progression, in 180 degree increments relative to the pel path direction. The pel path and line progression are constrained to be at right angles to one another.

4.6.5.1.2 Positioning. The position of a pel array element is defined by the PEL ARRAY REFERENCE POINT element. The reference point element affects the position of all pel array elements that follow it in the metafile, until it is overridden by a subsequent PEL ARRAY REFERENCE POINT ELEMENT. The

visual effect on whatever might already be positioned at or near a given reference point by the overlay of a pel array element is implementation-dependent.

4.6.5.1.3 Tiling. The tiling mechanism specified is based on the Tiled Raster Interchange Format work that has been developed relative to MIL-STD-1840 and ISO 8613 Part 7. The TILING MODE control element, when "on", indicates that all subsequent COMPRESSED PEL ARRAY elements are to be considered individual tiles within the tiled image until the a subsequent TILING MODE element sets the mode to "off". If the number of "tiles" defined while tiling mode is "on" is less than the number indicated by the TILED PEL ARRAY element, then the missing tiles are treated as encoded as "null background". The tiling offset parameter defines the position of the actual pel array within tile space, relative to the PEL ARRAY REFERENCE POINT. All tiles cover a portion of the pel array, the portions of the tile space outside of the tile array are artifacts of tiling and contain no information.

Page 19

Subclause 4.6.7: add the following after the subclause:

4.6.8 Conic Arc Element.

A Conic Arc is a bounded connected portion of a parent conic curve which consists of more than one point. The parent arc is either an ellipse, a parabola, or a hyperbola.

4.6.8.1 Parameterization. A conic arc is defined by the end points and the six parameters. The conic arc itself is defined by the six parameters in the following equation:

$$A(X^2) + BXY + C(Y^2) + DX + EY + F = 0$$

This parameterization assumes that VDC space is 4 quadrant cartesian coordinant space. The CONIC ARC TRANSFORMATION MATRIX element is then used to properly position the arc in the quadrant of VDC space defined by the VDC EXTENT element.

4.6.8.2 Geometric Concepts. The conic arc is defined by the start point, end point and the six parameters A-F. To determine the form of the conic arc, the quantities Q1, Q2 and Q3 are defined as follows:

$$Q1 = \text{determinant of } \begin{vmatrix} A & B/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{vmatrix}$$

$$Q2 = \text{determinant of } \begin{vmatrix} A & B/2 \\ B/2 & C \end{vmatrix}$$

$$Q3 = A + C$$

If $Q2 > 0$ and $(Q1 * Q3) < 0$, then the arc is elliptical;
if $Q2 < 0$ and $Q1 > 0$, then the arc is hyperbolic;
if $Q2 = 0$ and $Q1 > 0$, then the arc is parabolic.

In the case where the conic arc is elliptical, to distinguish the arc in question from its compliment, the direction of the arc with respect to VDC space must be from start point to end point in a counterclockwise direction.

In the case where the conic arc is parabolic or hyperbolic, the parameterization defines a unique portion of the parabola or a unique portion of a branch of the hyperbola, thus the direction is irrelevant.

4.6.9 Spline Curve Elements

The elements described in this section were derived from IGES V3.0, and are specialized for the case of two dimensions.

4.6.9.1 Parametric Spline Curve. The parametric spline curve is a sequence of parametric polynomial segments. The definition of this class of curves is generalized to allow for the representation of many different parametric spline curves using only one element. The following curve types have been assigned:

- 1: linear
- 2: quadratic
- 3: cubic
- 4: Wilson-Fowler
- 5: modified Wilson-Fowler
- 6: B spline

4.6.9.1.1 Parameterization. The degree of continuity parameter indicates the smoothness, or continuity of the curve with respect to arc length. The curve can either be continuous at all break points, continuous and have slope continuity at all break points, or be continuous and have both slope and curvature continuity at all break points.

The number of segments parameter is the number of polynomial segments to be used to define the curve. Each X,Y polynomial segment is evaluated using the eight polynomial coefficients associated with that segment (AX,BX,CX,DX,AY,BY,CY,DY). Each segment is delimited by its respective breakpoint.

4.6.9.1.2 Geometric Concepts. The following cubic polynomial equations will return the coordinates of the points of the i-th segment of the curve. Note that the coefficients D, or C and D will be zero if the polynomials are of degrees 2 or 1, respectively:

$$\begin{aligned} X(u) &= AX(i) + BX(i)(s) + CX(i)(s^2) + DX(i)(s^3) \\ Y(u) &= AY(i) + BY(i)(s) + CY(i)(s^2) + DY(i)(s^3) \end{aligned}$$

where $T(i) \leq u \leq T(i+1), i=1, \dots, N$ and $s = u - T(i)$.

The terminate point and derivatives are derived without computing the polynomials by evaluating the Nth polynomials and derivatives at $u = T(N+1)$. These data, divided by the appropriate factorial (i.e. the second derivative divided by 2!, the third by 3!), are used as the N+1 or terminate point values.

4.6.9.2 Rational B-Spline Curve.

4.6.9.2.1 Parameterization. The Rational B-Spline curve is parameterized where:

$$\begin{aligned} \text{start_param} &\leq t \leq \text{end_param}, \\ T(0) &\leq \text{start_param} < \text{end_param} \leq T(N) \end{aligned}$$

Thus for any parameter value t between $T(0)$ and $T(K+1)$, the sum of the basis functions satisfies the following identity:

$$b_0(t) + b_1(t) + \dots + b_K(t) = 1.$$

If all of the weights in the weight list are not equal, then the equation type is rational. Otherwise, if all of the weights are equal, then all of the weights cancel, the denominators sum to one and the equation type is polynomial.

4.6.9.2.2 Geometric Concepts. The parametric equation governing the definition of the rational B-spline curve is shown in the following expression:

$$\frac{W(0) \cdot P(0) \cdot b_0(t) + W(1) \cdot P(1) \cdot b_1(t) + \dots + W(K) \cdot P(K) \cdot b_K(t)}{W(0) \cdot b_0(t) + W(1) \cdot b_1(t) + \dots + W(K) \cdot b_K(t)}$$

where $W(i)$ are the weights, $P(i)$ are the control points and $b_i(t)$ are the basis functions. The basis functions are all non-negative piecewise polynomials determined by the degree and the knot sequence. The knot sequence is a non-decreasing list of real numbers $T(-M), \dots, T(0), \dots, T(K+1)$. Each basis function is supported for the knot sequence interval $[T(i-M), T(i+1)]$, where M is the degree of the basis function. Between any two adjacent knot values, the corresponding basis function can be expressed as a single polynomial. The basis functions are defined as follows:

Let $N(t; i-m, \dots, i+1)$ denote the B-Spline basis function of degree m supported on the interval $[t-m, t+1]$. The functions of degree d are defined with respect to those of $d-1$, as in the following:

$$N(t; i, \dots, i+d) = \frac{(t-t_0)N(t; i, \dots, i+d-1)}{t_d - t_0} + \dots + \frac{(t_d - t)N(t; i, \dots, i+d-1)}{t_d - t_1}$$

Since the denominators will be zero (0) in some cases, the convention $0/0 = 0$ is adopted for this definition.

4.X.X Compound Line

The compound line elements consist of the two elements, BEGIN COMPOUND LINE and END COMPOUND LINE. These elements permit the definition of a line that consists of a number of distinct elements, such as straight lines and arcs, which is treated as if it were a single line element. Thus, for example, line style would apply without change or interruption past a straight line segment onto a following arc segment. Likewise, the ends of the various component elements of the compound line are not treated as line end caps but rather as line joints.

4.X.X Compound Text Path

The compound text path elements consist of the two elements, BEGIN COMPOUND TEXT PATH and END COMPOUND TEXT PATH. It is functionally identical to Compound Line, except that it is used as a base line for text placement, rather than drawn by an interpreter.

The Compound Text Path permits arbitrary, complex placement of text. Each font symbol is placed with its reference point and alignment according to a tangent to the Compound Text Path. This implicit tangent is the logical base line for each character cell. If a symbol's reference point aligns with the junction of two line elements of the Compound Text Path, the logical base line is the line perpendicular to the perpendicular bisector of the tangents of both elements, passing through the reference point. Positioning of subsequent symbols is based upon the distance between symbols assuming a straight base line, but wrapped along the generalized curve of the Compound Text Path. If there is more text than path, the path for the excess text is the straight line described by the tangent to the last element of the Compound Text Path.

4.X.X Picture Composition

The picture composition elements consist of BEGIN CLIP REGION, END CLIP REGION, BEGIN SHIELD REGION, END SHIELD REGION, CLIP INDICATOR, and SHIELD INDICATOR.

The concepts of clip and shield regions are complementary. The clipping process discards everything that is visually outside the clip region whereas the shielding process discards everything that is inside the shield region. Whether clipping and shielding are in effect is determined by the respective settings of the CLIP INDICATOR and SHIELD INDICATOR (each is either ON or OFF).

Due to being able to define what amounts to closed figures for these regions, the clip region and shield regions can each provide a clipping and a shielding capability at the same time. For example, a "polygon with a hole" has an outer boundary and an inner boundary. The "filled area" of such a clipping region would be preserved with the area outside the "filled area" (including the contents of the hole) being removed from the picture. The shielding region has a complementary interpretation: the "filled area" itself is removed from the picture.

Note that the shielding effect of a clip region "hole" is independent of the SHIELD INDICATOR and likewise, the clipping effect of a shield region "hole" is independent of the CLIP INDICATOR.

Page 37

Subclause 4.7.7: replace the current text body with the following:

The CGM must be able to represent colours in a manner suitable to a broad spectrum of graphical devices and systems. Towards this goal, the CGM uses three colour models (RGB, CIE $L^*u^*v^*$, CYMK) and spot colour. The Metafile Descriptor element, COLOUR REPRESENTATION METHOD, allows metafile generators to specify which one is being employed.

The RGB additive colour model uses a 3-tuple of values providing the normalized weight of the red (R), green (G), and blue (B) components of the desired colour. This model is used in colour monitors, film writers, and input scanners.

The CIE $L^*u^*v^*$ uniform colour model uses a 3-tuple of values providing the normalized luminance (L^*), red-green chromaticness (u^*), and blue-yellow chromaticness (v^*) components of the desired colour. The advantages of the model over the others are (1) it separates chromaticness from luminance (allowing for easy monochrome display), (2) it is a uniform space for small colour differences (the Euclidean distance between two points in this model is more or less proportional to their perceived difference), and (3) it is an absolute colour model based colour matching experiments (thus being device independent and not requiring colour correction).

The CYMK subtractive colour model uses a 4-tuple of values providing the normalized weights of the cyan (C), yellow (Y), magenta (M), and black (K) components of the desired colour. This model is used by printers and Graphics Arts drum scanners. In theory, cyan, magenta, and yellow are meant to correspond to the red, green, and blue of the RGB model. In practice, actual inks used for colour printing only approximate this criterion. Black ink is added to attain a greater dynamic range than is possible with three colours alone. In particular, it allows the attainment of a much richer black than would be possible using only the first three inks.

Spot colour uses a character string representing a registered or private colour name (similar in format to named fonts). Use of the former is recommended for metafile transportability, because registration ensures unique naming of colours. Spot colours are registered in the ISO International Register of Graphical Items, which is maintained by the Registration Authority. When a spot colour has been approved by the ISO Working Group on Computer Graphics, the colour name will be assigned by the Registration Authority. Each registered colour will be specified in terms of its CIE $L^*u^*v^*$ coordinates.

The CGM provides two mechanisms for colour selection: 'direct' and 'indexed'. In 'direct' colour selection, the colour is specified by providing values for its normalized components (colour model) or by providing the character string defining its name (spot colour). (The term 'direct colour value' will refer to any direct colour specifier, and the term 'direct colour model value' will refer only to a direct colour specifier of a colour model (as opposed to spot colour)). In 'indexed' colour selection, the colour is specified by an index into a table of direct colour values. Selection of one of these mechanisms may be done by an element in each Picture Descriptor.

For 'indexed' colour selection, the COLOUR TABLE attribute element is provided for changing the contents of the colour table. This element may appear throughout the picture body. However, the effect of changes in the colour table on any existing graphical primitive elements that use the affected indices is not addressed in this Standard.

Direct colour model values are either a 3-tuple or 4-tuple of values providing the normalized weight of the desired colour components. In the abstract, each component is normalized to the continuous range of real numbers [0,1]. In a metafile, direct colour model value components are integers, and the Metafile Descriptor element, COLOUR VALUE EXTENT, allows metafile generators to specify the minimum and maximum metafile direct colour model values for normalization.

To address the problem of colour matching physical devices, the CGM provides a mechanism for colour calibration. The device space for a particular input/output device consists of the quantities that device uses for the measurement or the rendition of colour. Typical examples of devices and their associated device spaces are:

- colour monitors - this space is the intensities of the Red, Green, and Blue phosphors;
- colour film recorders - this space is the Red, Green, and Blue command values used for exposing the film;
- colour input scanners - this space is the values measured through the Red, Green, and Blue filters of the scanner;
- ink jet writers and thermal printers - this space is the ink values Cyan, Magenta, Yellow, and Black deposited onto paper;
- colour printing press - this space is the ink values which appear on separation films, which are later to be rendered on presses or proofing systems.

For all the kinds of devices listed above, it is important to appreciate that each of them has a well-defined meaning only in terms of the particular device with which it is used, but only an approximate meaning in terms of appearance. For example, if two pictures specified in terms of RGB are displayed on different monitors, the colours of the resulting images will not look identical. This is because no two monitor/display processor combinations are identical. If the two display systems of the same space are supplied by the same vendor, the images can be close enough for all but the most exacting applications; nonetheless, the difference does exist and must be taken into account. Similar considerations hold for the input scanner and ink spaces, with the extra problem that the viewing environment also affects the appearance.

Calibration is achieved by the imaging system converting a colour in the specified colour model to a colour in the reference colour model, and then converting to from the reference model to the device space for its imaging device. This applies for both input and output devices. Although this is conceptually two operations, it can be implemented as one transformation. These transformations require calibration data. The CGM does not allow for the interchange of calibration data. Instead, default values have been chosen to represent recognized standards for devices which use these models.

ANSI X3H3

Information Processing Systems --

Computer Graphics --

**Metafile for the Storage and Transfer
of Picture Description Information**

Part 1

**Functional Specification
(Clause 5)**

Addendum 3

Draft Document 1.3

Sub-clause 5.1: Table of data type abbreviations:

Replace

CD Colour Direct Three-tuple of non-negative real values for red, green, blue colour intensities.

with

MD Model Direct	Three-tuple or four-tuple of non-negative real values for colour definition within one of the three supported colour models.
CD Colour Direct	String or Model Direct (as determined by COLOUR REPRESENTATION METHOD).

Sub-clause 5.3.7: Replace the description as follows:

The precision for operands of data type model direct (MD) is specified for subsequent data of type MD. The precision is defined as the field width measured in units applicable to the specific encoding. Although the form of the parameter is encoding dependent, the parameter is a single specification that applies to each or all of the three or four components of parameters of type CM. The precisions of the individual components are not independently and differently specifiable by this element.

Sub-clause 5.3.10: Replace the description as follows:

The parameters represent an extent which bounds the direct colour model values that will be encountered in the metafile. It need not represent the exact extent of colour model values contained in the metafile.

Subclause 5.3: Add the following Metafile Descriptor Elements:

5.3.X COLOUR REPRESENTATION METHOD

Parameters:

colour representation method (one of: RGB, CIE $L^*u^*v^*$, CYMK, spot colour) (E)

Description:

Four methods of colour representation are supported: by one of three colour models (RGB, CIE $L^*u^*v^*$, CYMK) or by spot colour (names).

Only one colour representation method may be used within a metafile. The method may be defaulted or explicitly set with the COLOUR REPRESENTATION METHOD element. All occurrences of colour-setting elements (AUXILIARY COLOUR, LINE COLOUR, MARKER COLOUR, FILL COLOUR, EDGE COLOUR, TEXT COLOUR) as well as the colour lists of CELL ARRAY and PATTERN TABLE shall be in the current method. If used, COLOUR REPRESENTATION METHOD shall be in the Metafile Descriptor, after BEGIN METAFILE and before BEGIN METAFILE BODY.

References:
4.7.7

5.3.X FONT ATTRIBUTES

Parameters:

font number (IX)
number of attribute pairs (I)

List of: pairs of
Attribute type, Attribute value (S)

where attribute values must be one of:

typeface name (S)
family name (S)
typeface general class (IX)
typeface sub-class (IX)
typeface specific group (IX)
posture (one of: postures defined by ISO/IEC/DIS 9541-5, 6.6)
weight (one of: weights defined by ISO/IEC/DIS 9541-6, 6.7)
proportionate width (one of: see ISO/IEC/DIS 9541-5, 6.8)

Description:

The order of the attribute values indicates the relative importance of the attribute for font substitution. A missing value indicates no importance to be placed on the value. The following attribute types have been assigned:

typeface name
family name
typeface general class
typeface sub-class
typeface specific group
posture
weight
proportionate width

The font attributes provide a more detailed description of the fonts defined in the FONT LIST element so as to enable rational font matching in the event of the inability to exactly match a font from the font name specified in the FONT LIST element.

The font number will correspond to the font index defined in the FONT LIST element.

The typeface name is the name of the font typeface. Note that a typeface name implies particular values for the family name, weight, posture, and proportionate width.

The typeface general class is the most general grouping of fonts with similar characteristics. Typeface sub-classes are groupings that identifies the less general characteristics and starts to categorize typefaces into similar designs. Typeface specific groups are typeface groupings with very distinct and unique characteristics. Typefaces categorized to the typeface specific group level start to show similar characteristics that makes them reasonably eligible to be substituted for each other. The assigned font groups, and their attributes, are defined by the normative annex A of ISO 9541-5.

The posture of a font may be one of the following:

- 1: upright
- 2: obliquely slanted clockwise from the vertical, with no other design adjustments
- 3: obliquely slanted counter-clockwise from the vertical, with no other design adjustments
- 4: italic, slanted clockwise and the design adjusted for better appearance

The font weight is a measure of the boldness of the font. Assigned values are:

- 1: ultra light
- 2: extra light
- 3: light
- 4: semi light
- 5: medium
- 6: semi bold
- 7: bold
- 8: extra bold
- 9: ultra bold

The proportionate width is an indication of the ratio of character height to character width, and may be one of the following:

- 1: ultra condensed
- 2: extra condensed
- 3: condensed
- 4: semi condensed
- 5: medium
- 6: semi expanded
- 7: expanded
- 8: extra expanded
- 9: ultra expanded

In the preceding list, ultra condensed has the highest ratio of character height to character width, and ultra expanded has the lowest ratio of character height to character width.

References:

5.3.X FONTMETRIC DEFINITION

Parameters:

font index (IX)
 character index (C)
 left bearing (VDC)
 right bearing (VDC)
 character width (VDC)
 character height (VDC)
 offset from baseline (VDC)

Description:

The fontmetric information for each character used in each font specified is defined by this element. If this element is used, then the fontmetric data for each character used in the metafile must be specified. Characters not used by the metafile may also be specified, but are not required.

References:

5.3.X CHARACTER KERNING MODE

Parameters:

character kerning mode (one of: none, pair, sector, track) (E)

Description:

Defines the kerning style, if any, for the metafile.

References:

5.3.X CHARACTER KERNING TABLE

Parameters:

To be determined.

Description:

The data defined by this element will be dependant upon which, if any, kerning styles are supported. In general, however, the information will be that which is required to kern characters.

References:

5.3.X BEGIN GLYPH DEFINITION

Parameters:

character index (IX)

Description:

BEGIN GLYPH DEFINITION delimits the beginning of a definition of a entity that will be treated as a single "compound primitive". The elements that make up the compound figure can be any combination of non-closed line elements such as POLYLINE, CIRCULAR ARC 3 POINT, CIRCULAR ARC CENTRE, ELLIPTICAL ARC, <new curve elements>, BEGIN REQUIRED FEATURE, END REQUIRED FEATURE, and CLOSE FEATURE.

The figure defined by the component line primitives will form a character glyph, and will be assigned to the character code referenced by the value of the character index parameter.

The interior of the glyph (see 4.6.4.4) is filled according to the current filled-area attributes. The composite figure will be either a single figure or a set of figures, which is filled according to the parity (odd or even) algorithm described under the POLYGON element, with the exception that the transition from a vertex marked as a 'closure vertex' to the next point specified in the definition does not constitute a boundary to the fill algorithm. A vertex is marked as a 'closure vertex' by the use of the CLOSE CONTOUR element.

The individual figures of the set are not filled individually. The figures in the set may be disjoint (as in the 'dot' in the letter 'i'), may create 'holes' (as in the interior of the letter 'o'), or may overlap.

References:

5.3.X END GLYPH DEFINITION

Parameters:

None

Description:

END GLYPH DEFINITION delimits the end of a compound outline definition of a character glyph.

References:

5.3.X BEGIN REQUIRED FEATURE

Parameters:

None

Description:

BEGIN REQUIRED FEATURE delimits the beginning of a definition of a entity that will be treated as a single "compound primitive". The elements that make up the compound figure can be any combination of non-closed line elements such as POLYLINE, CIRCULAR ARC 3 POINT, CIRCULAR ARC CENTRE, ELLIPTICAL ARC, CLOSE CONTOUR, and <new curve elements>. The figure may be a disjoint figure.

If used, BEGIN REQUIRED FEATURE shall occur within the scope of a BEGIN GLYPH DEFINITION block. The resulting composite figure will be a specific feature of the character glyph being defined which is required for proper rendering of the glyph (such as a serif).

References:

5.3.X END REQUIRED FEATURE

Parameters:

None

Description:

END REQUIRED FEATURE delimits the end of a compound outline definition of a specific feature of a character glyph required for proper rendering of the glyph.

References:

5.3.X CLOSE CONTOUR

Parameters:

None

Description:

CLOSE CONTOUR delimits the end of a compound outline of a character glyph. The last point specified is marked as a 'closure vertex'.

References:

Page 55

Sub-clause 5.4.2, first paragraph of description:

Change "red, green, and blue" into "direct"

Page 57

Sub-clause 5.4.7, first line of second paragraph of description:

Change "RGB" into "a direct colour value"

Page 60

Subclause 5.5: Add the following Control Elements:

5.5.X BEGIN COMPOUND LINE

Parameters:

None

Description:

BEGIN COMPOUND LINE delimits the beginning of a definition of a entity that will have consistent line attributes and will be treated as a single "compound primitive". The elements that make up the compound line can be any combination of non-closed line elements such as POLYLINE, CIRCULAR ARC 3 POINT, CIRCULAR ARC CENTRE, ELLIPTICAL ARC, or <new curve elements>.

References:

5.5.X END COMPOUND LINE

Parameters:

None

Description:

END COMPOUND LINE delimits the end of a compound line definition.

References:

5.5.X BEGIN TEXT PATH

Parameters:

None

Description:

BEGIN TEXT PATH delimits the beginning of a definition of a entity that will provide the path in which a text string will be drawn. The elements that make up the compound text path can be any combination of non-closed line elements such as POLYLINE, DISJOINT POLYLINE,

CIRCULAR ARC 3 POINT, CIRCULAR ARC CENTRE, ELLIPTICAL ARC, or <new curve elements>.

Once defined, the compound text path takes the place of the text path as defined by the TEXT PATH element and the CHARACTER ORIENTATION elements. The skew of the characters is still relative to that specified in the CHARACTER ORIENTATION element, but the placement of subsequent characters is along the compound text path instead of in a line along the character up vector or character base vector.

References:

5.5.X END TEXT PATH

Parameters:

None

Description:

END TEXT PATH delimits the end of a compound text path definition.

References:

5.5.X BEGIN CLIP REGION

Parameters:

None

Description:

BEGIN CLIP REGION delimits the beginning of a definition of a entity that will provide the clipping region. When CLIP INDICATOR is 'on' only the portions of graphics elements inside or on the boundary of the clipping region are drawn. The elements that make up the clipping region can be any combination of closed or non-closed elements such as POLYLINE, DISJOINT POLYLINE, POLYGON, POLYGON SET, CIRCULAR ARC 3 POINT, CIRCULAR ARC 3 POINT CLOSE, CIRCULAR ARC CENTRE, CIRCULAR ARC CENTRE CLOSE, ELLIPTICAL ARC CLOSE, or <new curve elements>. The entity thus defined is essentially a closed figure whose boundary is used as the clipping boundary.

Once defined, the clipping region takes the place of the clipping region defined in CLIP RECTANGLE.

References:

5.5.X END CLIP REGION

Parameters:

None

Description:

END CLIP REGION delimits the end of a clipping region definition.

References:

5.5.X BEGIN SHIELD REGION

Parameters:

None

Description:

BEGIN SHIELD REGION delimits the beginning of a definition of a entity that will provide the shielding region. When SHIELD INDICATOR is 'on' only the portions of graphics elements outside of the shielding region are drawn. The elements that make up the shielding region can be any combination of closed or non-closed elements such as POLYLINE, DISJOINT POLYLINE, POLYGON, POLYGON SET, CIRCULAR ARC 3 POINT, CIRCULAR ARC 3 POINT CLOSE, CIRCULAR ARC CENTRE, CIRCULAR ARC CENTRE CLOSE, ELLIPTICAL ARC CLOSE, or <new curve elements>. The entity thus defined is essentially a closed figure whose boundary is used as the shielding boundary.

References:

5.5.X END SHIELD REGION

Parameters:

None

Description:

END SHIELD REGION delimits the end of a shielding region definition.

References:

5.5.X SHIELDING INDICATOR

Parameters:

shield indicator (one of: off, on) (E)

Description:

When SHIELD INDICATOR is 'off', shielding of graphical primitive elements is not required.

When SHIELD INDICATOR is 'on', only those portions of graphical primitive elements outside of the shielding region are drawn.

References:

5.5.X TILING MODE

Parameters:

tiling mode (one of: off,on) (E)

Description:

When TILING MODE is 'off', subsequent COMPRESSED PEL ARRAY elements are to be treated as independent images displayed per the corresponding PEL ARRAY ORIENTATION and PEL ARRAY DIMENSIONS attribute elements, and positioned in VDC space via the PEL ARRAY REFERENCE POINT element.

When the TILING MODE is "on", each subsequent COMPRESSED PEL ARRAY element is treated as a single tile in a tiled image and indexed by the Pel Array Identifier parameter. The PEL ARRAY ORIENTATION and PEL ARRAY DIMENSIONS elements describe the sizing and orientation of the entire tiled image. The dimensional information for the tiles in the image is contained in the TILED PEL ARRAY element.

References:

Page 68

Sub-clause 5.6.9: Add the following at the end of the third paragraph of the description:

Note that COLOUR PRECISION only applies to direct colour model values.

Page 77

Subclause 5.6: Add the following Graphical Primitive Elements:

5.6.X CONIC ARC

Parameters:

start point (P)
end point (p)
A,B,C,D,E,F (6R)

Description:

A conic arc is drawn which is defined as follows:

A conic arc is defined by the end points and the six parameters. The conic arc itself is defined by the six parameters in the following equation:

$$A(X^2) + BX_vY_v + C(Y^2) + DX_v + EY_v + F = 0$$

where (X_v, Y_v) are the horizontal and vertical axes, respectively, of VDC space. In order for the conic arc to be processed correctly by the receiving system given the above representation, the conic arc entry must be positioned such that each of its axes is parallel to either the X_v axis or Y_v axis. The arc is then positioned correctly in VDC space by using the value of the CONIC ARC TRANSFORMATION MATRIX element.

To determine the form of the conic arc, the quantities Q1, Q2 and Q3 are defined as follows:

$$Q1 = \text{determinant of } \begin{vmatrix} A & B/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{vmatrix}$$

$$Q2 = \text{determinant of } \begin{vmatrix} A & B/2 \\ B/2 & C \end{vmatrix}$$

$$Q3 = A + C$$

If $Q2 > 0$ and $(Q1 * Q3) < 0$, then the arc is an ellipse.
If $Q2 < 0$ and $Q1 > 0$, then the arc is a hyperbola.
If $Q2 = 0$ and $Q1 > 0$, then the arc is a parabola.

In the case where the conic arc is elliptical, to distinguish the arc in question from its complement, the direction of the arc with respect to the definition space must be from start point to end point in a counterclockwise direction.

In the case where the conic arc is parabolic or hyperbolic, the parameterization defines a unique portion of the parabola or a unique portion of a branch of the hyperbola, thus the direction is irrelevant.

The direction of the conic arc with respect to VDC space is determined by the original direction of the arc in definition space, in conjunction with the action of the CONIC ARC TRANSFORMATION MATRIX element.

References:

4.

5.6.X CONIC ARC TRANSFORMATION MATRIX

Parameters:

matrix elements

if the VDC type is 'integer',
R11,R12,R13,R21,R22,R23 (6I)

if the VDC type is 'real',
R11,R12,R13,R21,R22,R23 (6R)

Description:

This element is intended to work in conjunction with the CONIC ARC element to transform the conic arc to its final position and orientation in VDC space. The Transformation Matrix entity transforms the defining point coordinates by means of a matrix multiplication. This transformation is achieved by applying the matrix multiplication to the coordinates of the conic arc.

The notation for this transformation is as follows:

$$\begin{bmatrix} R11 & R12 & R13 \\ R21 & R22 & R23 \end{bmatrix} \begin{bmatrix} X_{in} \\ Y_{in} \end{bmatrix} = \begin{bmatrix} X_{out} \\ Y_{out} \end{bmatrix}$$

where $|Rij|$ is the transformation matrix, (X_{in}, Y_{in}) is the coordinate to be transformed, and (X_{out}, Y_{out}) is the coordinate resulting from the transformation. Both the input and output coordinate systems are assumed to be orthogonal, cartesian and right-handed.

References:

4.

5.6.X PARAMETRIC SPLINE CURVE

Parameters:

curve_type (DC)
H-degree of continuity (I)
N-number of segments (I)
T-break point list for polynomial ((N+1)R)
X coordinate polynomial list (N sets of four)

AX,BX,CX,DX ((N*4)R)

Y coordinate polynomial list (N sets of four)

AY,BY,CY,DY ((N*4)R)

Description:

The parametric curve to be drawn is defined as follows:

The parametric spline curve is a sequence of parametric polynomial segments. The parameterization shown above is generalized to allow for the representation of many different parametric spline curves using this one element. The curve_type parameter indicates the type of parametric curve as it was represented in the sending system before being converted to this generic form. The following curve types have been assigned:

- 1: linear
- 2: quadratic
- 3: cubic
- 4: Wilson-Fowler
- 5: modified Wilson-Fowler
- 6: B-Spline

Values above 6 are reserved for registration and future standardization, and negative values are available for implementation-dependent use.

The degree of continuity parameter, H, indicates the smoothness, or continuity of the curve with respect to arc length. If H=0, the curve is continuous at all break points. If H=1, the curve is continuous and has slope continuity at all break points. If H=2, the curve is continuous and has both slope and curvature continuity at all break points.

The number of segments parameter, N, is the number of polynomial segments to be used to define the curve. Each segment is defined by a cubic polynomial in X and Y that is evaluated using the eight polynomial coefficients associated with that segment; Ax,Bx,Cx,Dx,Ay,By,Cy,Dy. Segment i is delimited by its knots, T(i) and T(i+1).

The parametric spline curve is displayed with the current line attributes.

References:

4.

5.6.X RATIONAL B-SPLINE CURVE

Parameters:

K-upper index of sum (I)
M-degree of the basis function (I)
equation_type flag (one of: rational, polynomial) (E)
T-knot sequence list ((K+M+1)R)
W-weight list ((K+1)R)
control point list ((K+1)P)
start_param,end_param (2R)

Description:

A two dimensional rational B-spline curve is drawn. The realization of this element is based upon the Rational B-Spline Curve entity of IGES V3.0. The specifications are identical except that the curve is constrained to two dimensions, i.e. the Z Polynomial is assumed to be zero (0).

Valid values of the upper index and degree parameters are non-negative integers.

Valid values of the control points are such that no two adjacent points are coincident.

If all of the weights in the weight list *W* are not equal, then the *equation_type* flag is set to Rational, otherwise the *equation_type* flag is set to Polynomial.

References:

4.

5.6.X COMPRESSED PEL ARRAY

Parameters:

PID-pel array identifier (I)
T-encoding type (one of: T4, T6, LZW, Bitmap, null background, null foreground) (E)
P-pel path (one of: 0, 90, 180, 270) (E)
L-line progression (one of: 90, 270) (E)
S-pel spacing (VDC)
spacing_ratio (R)
N-number of pels per line (I)
NL-number of lines (I)
pel array (BS)

Description:

A compressed pel array image is defined as follows:

The pel array identifier, PID, is used as the tiling index when Tiling Mode is ON. When the Tiling Mode is OFF, the PID is used as an identifier to uniquely label the pel array for manipulation or access purposes.

The encoding type parameter, T, specifies the compression format used to encode the image. If T is specified as "T4", the image is encoded according to the one or two dimensional scheme defined in CCITT Recommendation T.4 (Group 3 facsimile). If T is "T6", the image is encoded according to the two dimensional scheme defined in CCITT Recommendation T.6 (Group 4 facsimile). T specified as "null background" or "null foreground" is used mainly when in tiling mode to indicate that all pels in the tile are known to be background or foreground and the tile has no encoded content. If either of these are specified, then the Pel Array itself will be a zero length or "null" binary string.

The pel path parameter, P, is the direction of progression of successive pels along a line relative to the VDC X axis. This parameter, in conjunction with the pel spacing, S, and the number of pels per line, N, implicitly define the line position, length and granularity for each line in the decoded pel array. The spacing_ratio is defined as the ratio M/S, where M is the distance in VDC units between two successive lines of pels, and S is the Pel Spacing, which is the distance in VDC between two adjacent pels in a line. Note that even if the VDC TYPE is Integer, the values of the Pel Spacing S and the line spacing M will be of type Real.

The line progression parameter, L, is the direction of progression of successive of pel lines and is expressed as a direction relative to P. L, in conjunction with the spacing_ratio and the number of lines, NL, implicitly defines the size of the decoded image in the direction of L. The line spacing (LS) of the lines of pels can be determined as follows:

$$LS = \text{spacing_ratio} * S$$

The pel array itself is stored in either of the formats defined by T, encoded as a bit_stream.

References:

4.

5.6.X TILED PEL ARRAY

Parameters:

TID-tiled pel array ID (I)
TD-tiled pel array dimensions (2I)
NP-number of pels per tile line (I)
NL-number of lines per tile (I)
TO-tiling offset (2I)

Description:

A tiled pel array image is defined as follows:

The tiled pel array, TID, uniquely identifies the tiled image. Valid values of this ID are positive integers.

The tiled pel array dimensions parameter, TD, consists of two positive integers corresponding to the number of tiles in the direction of the Pel Path and Line Progression parameters, respectively, of the PEL ARRAY ORIENTATION element. Multiplying the two TD integers together will give the total number of tiles contained in the tiled image.

The number of pels per tile line parameter, NP, specifies the tile dimension in units of Pel Spacing found in the PEL ARRAY DIMENSIONS element, in the Pel Path direction of the PEL ARRAY ORIENTATION element, for each tile of the tiled image.

The number of lines per tile parameter, NL, specifies the tile dimension in units of line spaces, derived from the Spacing Ratio of the PEL ARRAY DIMENSIONS element, in the Line Progression direction of the PEL ARRAY ORIENTATION element, for each tile of the tiled image.

The tiling offset parameter, TO, specifies the location of the pel array image within tiling space by defining the offset of the first pel of the pel array from the PEL ARRAY REFERENCE POINT. The offset is specified by two positive integers corresponding to the number of pel spaces in the Pel Path direction and line spaces in the Line Progression direction, respectively.

References:

5.6.X IMAGE APERTURE

Parameters:

X1,Y1,X2,Y2 (4I)

Description:

The element defines the rectangular area of pels in the decoded pel array that is to appear in the metafile picture. This element should immediately precede the Pel Array element to which it applies, as this element will affect all subsequent Pel Array elements until another Image Aperture is explicitly defined.

The four integers form two coordinate pairs, (X1,Y1) and (X2,Y2) corresponding to the first and last pels to appear, respectively, where $X1 \leq X2$ and $Y1 \leq Y2$. For example, (6,2) would specify the seventh pel in line 3, given that (0,0) specifies the first pel on the first line.

These two corner pels, (X1,Y1) and (X2,Y2), define the rectangular portion of interest of the Pel Array image to be positioned in the metafile picture. The first pel defined by (X1,Y1) above is to be positioned at the reference point of the Pel Array element, with the remainder of the image realized as per the Pel Spacing, Spacing Ratio, Pel Path, and Line Progression parameters of the Pel Array element. Valid values of the IMAGE APERTURE parameters are non-negative integers.

References:

4.

5.6.X PEL ARRAY REFERENCE POINT

Parameters:

reference_point (P)

Description:

The pel array reference point defines the position of the upper left-hand corner of the pel array element to be displayed. If the pel path and line progression are thought of as vectors, the upper left-hand corner is defined as point of origin for the two vectors.

References:

4.

5.6.X BOXED TEXT

Parameters:

two points (2P)
flag (one of: not final, final) (E)
string (S)

Description:

The two points specified represent diagonally opposite corners of a rectangle oriented parallel to the VDC axes. BOXED TEXT behaves as does TEXT, with the exception that the text is constrained to be within the rectangle defined by the two points.

The character codes specified in the string are interpreted to obtain the associated symbols from the currently selected character set. Characters are displayed on the view surface as specified by the text attributes. Format effector control characters (such as CR, LF, BS, HT, VT, and FF) are permitted in a string but their interpretation is implementation dependent. Control characters used for character set invocation and designation (SI, SO, ESC, SS2, and SS3) are permitted according to the setting of CHARACTER CODING ANNOUNCER.

The characters are dimensioned according to the height and width of the bounding box and are oriented according to CHARACTER ORIENTATION. The direction of character placement in the string relative to CHARACTER ORIENTATION is according to TEXT PATH.

All text in the string is displayed within the specified bounding box. If necessary, the values of the text attributes CHARACTER HEIGHT, TEXT PRECISION, and TEXT FONT INDEX which are used to display the string are varied to force the text to completely fill the bounding box.

The flag parameter is used to permit changing the following text attributes and control elements within a string which will be aligned as a single block: TEXT FONT INDEX, TEXT PRECISION, CHARACTER EXPANSION FACTOR, CHARACTER SPACING, TEXT

COLOUR, CHARACTER HEIGHT, CHARACTER SET INDEX, ALTERNATE CHARACTER SET INDEX, TEXT BUNDLE INDEX, AUXILIARY COLOUR, and TRANSPARENCY.

If the flag is set to 'not final', the character codes in the string parameter are accumulated, along with the current attribute settings. In this case, only the attribute setting elements listed above are allowed between this element and the APPEND TEXT element. With the exception of the ESCAPE, no other metafile elements of any type are allowed. ESCAPE is permitted but has no standardized effect.

If the flag is set to 'final', the string parameter constitutes the entire string to be displayed. It is this complete string to which the text bound box applies. The position of the string is determined by the bounding box. Text elements with a null string parameter are legal and may be followed by the allowed text attributes and APPEND TEXT as described above.

NOTE- TEXT PRECISION is included in the attributes which may be changed to achieve the text restriction because TEXT PRECISION controls the relationship between currently set values of text attributes and the values actually used for display of a string (the "realized" values). The realization of the text restriction required by the BOXED TEXT elements may mandate another mapping from requested to realized attribute values than would be allowable under the current TEXT PRECISION. Hence the requirements of the current TEXT PRECISION may have to be ignored to achieve the proper display of the BOXED TEXT element.

When the flag is 'not final' this element causes a state transition in the state diagram of figure 12, into the PARTIAL TEXT state.

References:

4.

5.6.X PATH TEXT

Parameters:

flag (one of: not final, final) (E)
string (S)

Description:

The PATH TEXT element shall be immediately preceded by a compound line definition. The alignment point of the string will be a point along the defined text path corresponding to the current TEXT ALIGNMENT value.

The character codes specified in the string are interpreted to obtain the associated symbols from the currently selected character set. Characters are displayed on the view surface as specified by the text attributes. Format effector control characters (such as CR, LF, BS, HT, VT, and FF) are permitted in a string but their interpretation is implementation dependent. Control characters used for character set invocation and designation (SI, SO, ESC, SS2, and SS3) are permitted according to the setting of CHARACTER CODING ANNOUNCER.

The characters are dimensioned according to the CHARACTER HEIGHT and CHARACTER EXPANSION FACTOR and are oriented according to CHARACTER ORIENTATION. The direction of character placement in the string relative to CHARACTER ORIENTATION is along the path defined within the scope of the preceding BEGIN TEXT PATH and END TEXT PATH elements. If the string length exceeds the length of the path, the characters of the string will continue to be placed along the path defined by a vector whose tail is the last point of the path and whose direction is the direction of the path at the last point.

The flag parameter is used to permit changing the following text attributes and control elements within a string which will be aligned as a single block: TEXT FONT INDEX, TEXT

PRECISION, CHARACTER EXPANSION FACTOR, CHARACTER SPACING, TEXT COLOUR, CHARACTER HEIGHT, CHARACTER SET INDEX, ALTERNATE CHARACTER SET INDEX, TEXT BUNDLE INDEX, AUXILIARY COLOUR, and TRANSPARENCY.

If the flag is set to 'not final', the character codes in the string parameter are accumulated, along with the current attribute settings. In this case, only the attribute setting elements listed above are allowed between this element and the APPEND TEXT element. With the exception of the ESCAPE, no other metafile elements of any type are allowed. ESCAPE is permitted but has no standardized effect.

If the flag is set to 'final', the string parameter constitutes the entire string to be displayed. It is this complete string to which the text bounding box applies. The position of the string is determined by the bounding box. Text elements with a null string parameter are legal and may be followed by the allowed text attributes and APPEND TEXT as described above.

NOTE- When the flag is 'not final' this element causes a state transition in the state diagram of figure 12, into the PARTIAL TEXT state.

References:

4.

5.6.X RESTRICTED PATH TEXT

Parameters:

flag (one of: not final, final) (E)
string (S)

Description:

The RESTRICTED PATH TEXT behaves as does PATH TEXT, with the exception that the text is constrained to be the same length as the path along which it is to be drawn.

The character codes specified in the string are interpreted to obtain the associated symbols from the currently selected character set. Characters are displayed on the view surface as specified by the text attributes. Format effector control characters (such as CR, LF, BS, HT, VT, and FF) are permitted in a string but their interpretation is implementation dependent. Control characters used for character set invocation and designation (SI, SO, ESC, SS2, and SS3) are permitted according to the setting of CHARACTER CODING ANNOUNCER.

The characters are dimensioned according to the CHARACTER HEIGHT and CHARACTER EXPANSION FACTOR and are oriented according to CHARACTER ORIENTATION. The direction of character placement in the string relative to CHARACTER ORIENTATION is along the path defined within the scope of the preceding BEGIN TEXT PATH and END TEXT PATH elements.

All text in the string is displayed along the specified COMPOSITE TEXT PATH. If necessary, and only if necessary, the values of the text attributes CHARACTER HEIGHT, CHARACTER EXPANSION FACTOR, CHARACTER SPACING, TEXT PRECISION, and TEXT FONT INDEX which are used to display the string are varied to force the text to be the same length as the defining composite path. It is only the realized values of these attributes, used to display this single string, which are varied. The method of varying the attributes is implementation dependent.

The flag parameter is used to permit changing the following text attributes and control elements within a string which will be aligned as a single block: TEXT FONT INDEX, TEXT PRECISION, CHARACTER EXPANSION FACTOR, CHARACTER SPACING, TEXT COLOUR, CHARACTER HEIGHT, CHARACTER SET INDEX, ALTERNATE CHARACTER SET INDEX, TEXT BUNDLE INDEX, AUXILIARY COLOUR, and TRANSPARENCY.

If the flag is set to 'not final', the character codes in the string parameter are accumulated, along with the current attribute settings. In this case, only the attribute setting elements listed above are allowed between this element and the APPEND TEXT element. With the exception of the ESCAPE, no other metafile elements of any type are allowed. ESCAPE is permitted but has no standardized effect.

If the flag is set to 'final', the string parameter constitutes the entire string to be displayed. It is this complete string to which the text bound box applies. The position of the string is determined by the bounding box. Text elements with a null string parameter are legal and may be followed by the allowed text attributes and APPEND TEXT as described above.

NOTE- TEXT PRECISION is included in the attributes which may be changed to achieve the text restriction because TEXT PRECISION controls the relationship between currently set values of text attributes and the values actually used for display of a string (the "realized" values). The realization of the text restriction required by the RESTRICTED PATH TEXT elements may mandate another mapping from requested to realized attribute values than would be allowable under the current TEXT PRECISION. Hence the requirements of the current TEXT PRECISION may have to be ignored to achieve the proper display of the RESTRICTED PATH TEXT element.

When the flag is 'not final' this element causes a state transition in the state diagram of figure 12, into the PARTIAL TEXT state.

References:

4.

Page 95

Sub-clause 5.7.32: Add the following at the end of the third paragraph of the description:

Note that COLOUR PRECISION only applies to direct colour model values.

Page 97

Subclause 5.7: Add the following attribute elements:

5.7.X LINE TYPE DEFINITION

Parameters:

linetype (IX)
dash unit selector (one of: VDC, mm, native device units, abstract) (E)
dash repeat length (R)
adaptive flag (one of: no, yes) (E)
list of dash elements (nl)

Description:

This element defines a linetype and associates it with an index for future reference. Parameter 'linetype' is the index of linetype being defined. The parameter 'list of dash elements' is the definition to be associated with the index. The first element is a dash, second a space, etc. - the defined linetype is solid for I1 units, gap for I2 units, solid for I3 units, etc. N must be positive, and each dash element (I) non-negative. N=1 means a solid line; I=0 interpreted as a dot.

The units of the 'dash repeat length' are specified by the 'dash unit selector' parameter. The value of 'abstract' indicates that the implementation may normalize and map the sum of the dash pattern

elements at its discretion. The 'dash repeat length' defines the length of one complete cycle of the dash pattern, measured in the units of 'dash unit selector'.

An "adaptive" linetype is one where every vertex falls on an inked portion of the line. This is accomplished in plotters by temporarily modifying the duty cycle for each line segment (ceiling function) such that there is always an integral number of repeats (and all predefined linetypes have their gaps_array defined such that they begin and end with inked or "pen down" portions).

References:

5.7.X HATCH STYLE DEFINITION

Parameters:

hatch index (IX)
style indicator (one of: parallel, crosshatch) (E)
hatch space units selector (one of: VDC, mm, device units, abstract) (E)
angle (2R)
duty cycle length (R)
list of hatch elements (nI)

Description:

This element defines a hatch style and associates it with an index for future reference.

The 'hatch index' parameter defines the index of hatch style being defined. The 'list of hatch elements' is an array that defines alternating line width and gap width -- i.e., the width of a hatch line followed by the width of the space to the next hatch line. The center of the first hatch line is matched up with PATTERN REFERENCE POINT, if implemented. 0 interpreted as thinnest line width available.

The 'hatch space units selector' specifies the units of 'duty cycle length'. It also controls the manner of transformation of the hatching: If VDC, then the hatching transforms with segment transform and anisotropic transforms (as if hatching had done POLYLINES); otherwise, the hatching is like "wallpaper" that shows through the polygon-shaped hole -- everything is defined in device units and hatching performed in device space. The value of 'abstract' indicates that the implementation may normalize and map the sum of the list of hatch elements at its discretion. The 'duty cycle length' is measured perpendicular to the hatch line. The sum of hatch elements in the hatch element list is normalized to this distance before presentation of the hatch on the view surface.

The 'angle' parameter is measured in the units specified by the 'hatch space units selector'. It consists of two components, dx and dy, defining a vector.

References:

5.7.X LINE CAP

Parameters:

line cap indicator (one of: butt, round, projecting square) (E)

Description:

The line cap style is defined for subsequent line elements. The line cap style determines the appearance of open endpoints (as opposed to interior vertices) of line elements. The defined styles are:

butt cap:	the line is squared off at the endpoint, there is no projection beyond the endpoint.
round cap:	a semicircular arc with diameter equal to the line width is drawn around the endpoint and filled in. The drawn line thus projects beyond the endpoint.
projecting square cap:	the line is squared off at a distance equal to half the line width beyond the endpoint.

References:

5.7.X LINE JOIN

Parameters:

line join indicator (one of: miter, round, bevel) (E)

Description:

The line join style is defined for subsequent line elements. The line join style defines the appearance of interior vertices of polyline elements and of compound line elements. The defined styles are:

miter join:	the outer edges of the two adjoining line segments are extended until they meet at a point.
round join:	a circular arc with diameter equal to the line width is drawn around the vertex between the adjoining segments and is filled in, producing a rounded corner.
bevel join:	the adjoining line segments are terminated with a butt cap, and the resulting triangular notch is filled in.

References:

5.7.X EDGE CAP

Parameters:

edge cap indicator (one of: butt, round, projecting square) (E)

Description:

The edge cap style is defined for subsequent edge elements. The edge cap style determines the appearance of open endpoints of filled area edges (such as may result from a mixture of visible and invisible edge segments). The defined styles are:

butt cap:	the edge is squared off at the vertex, there is no projection beyond the endpoint.
round cap:	a semicircular arc with diameter equal to the edge width is drawn around the endpoint and filled in. The drawn edge thus projects beyond the endpoint.
projecting square cap:	the edge is squared off at a distance equal to half the edge width beyond the endpoint.

5.7.X FONT SCORE TYPE

Parameters:

score type (one of: `strikeout`, `underline`, `overscore`, or `none`)

Description:

The score type is set to the value specified by the parameter.

The following score types are assigned:

- 1: `strikeout`
- 2: `underscore`
- 3: `overscore`
- 4: `none`

The score type '`strikeout`' is a line drawn through each character in the string at the 'half' vertical alignment position.

The score type '`underscore`' is a line drawn below each character in the string at the 'bottom' vertical alignment position.

The score type '`overscore`' is a line drawn above each character in the string at the 'top' vertical alignment position.

The score type '`none`' removes all scoring.

References:

ANSI X3H3

Information Processing Systems --

Computer Graphics --

Metafile for the Storage and Transfer
of Picture Description Information

Part 1

Functional Specification

(Clause 6)

Addendum 3

Draft Document 2.0

Page 100

Clause 6: Add the following default specifications:

CONIC ARC TRANSFORMATION MATRIX:	identity matrix
PEL ARRAY CLIP RECTANGLE:	(0,0) upper left, (N-1,L-1) lower right, where N is the number of pels per line, and L is the number of lines in the last COMPRESSED PEL ARRAY element
PEL ARRAY REFERENCE POINT:	upper left-hand corner point of the default VDC extent
Clipping Region: (BEGIN/END CLIP REGION)	VDC Extent
Shielding Region: (BEGIN/END SHIELD REGION)	VDC Extent
CLIP INDICATOR:	On
SHIELD INDICATOR:	Off
COLOUR REPRESENTATION METHOD:	RGB

Attachment 7:

US Comments on Addendum 3 Working Draft

US Comments CGM Addendum 3 Working Draft

1. Introduction

This document contains the US comments on the Working Draft of Addendum 3, document version dated 89/07/18. The comments are divided into three sections: the overall comments of the next section; a checklist comparing Add.3 Working Draft content to the NWI and requirements documents; and a section of detailed technical comments. In addition the US intends to bring contributions of revised text to the Olinda Metafile RG Meeting.

2. Overall Comments

2.1 Schedule

The schedule must be considered a requirement at least as strong as any of the other requirements. This may force compromises on technical content. In particular attempts to develop technical areas which will likely be lengthy, or will require liaison with other standards work which has a significantly longer timetable, should be postponed. The US believes that the work can and should be completed on the timetable projected in the NWI.

2.2 Completeness of the draft

The US believes that the current Working Draft is not sufficiently complete for DP registration. The functionality is incomplete with respect to that detailed in the NWI and Requirements documents, as pointed out below in "NWI/Requirements Checklist." The defaults are missing for many elements that need them. With the correction of these omissions, and the addition of the encodings, the Working Draft will be complete enough for DP registration and ballot.

2.3 Compatibility with other SC24 Work

As stated in the NWI the Addendum 3 work should coordinate with the SC24 work in APIs and Reference Models. However the Add.3 metafile has strong ties to common practice which is outside of the scope of SC24 standards as well — proprietary CAD databases, desktop publishing systems, page description languages, and graphic arts workstations to name a few. The US believes that there should not be arbitrary differences between this work and work on existing or new APIs. However the US defines the following priorities for possibly resolving conflicting requirements: schedule and identified functional requirements must have the highest priority.

2.4 Document form

Addendum 3 should ultimately be processed as a complete document, i.e., CGM plus Addendum 1 plus Addendum 3 should be merged into a single document for review and advancement. Because this is going to be fairly unwieldy, we think that this should be done at the stage that DP text is prepared and the DP ballot initiated. At the Working Draft stage it is easier to focus on functional completeness and correctness (and ignore document completeness and consistency) in the current

format. If two DPs are anticipated, then the consolidation should happen at the second DP stage.

2.5 Relationship to Addendum 1

Addendum 3 should be a proper superset of Addendum 1. A legal Add.1 metafile should be a legal Add.3 metafile.

2.6 Further Coordination with Improved Text Model Study Group

The US recommends that any further work on Improved Text Model should take place in the Reference Models RG of WG1. In the meantime the work done so far, and useful extensions from current practice in the industry, should guide the improved text facilities of Add.3.

2.7 Further Coordination with PDG Study Group

No specific requirements for Add.3 have derived from the work of the PDG study group. The US believes that such requirements are most likely to be applicable to Addendum 2, the 3D metafile, when and if the scope of such is defined. If any specific requirements for 2D metafile support can be stated, then those should be observed in the work on Add.3.

3. NWI/Requirements Checklist

The following list is extracted from the NWI and the requirements document for Addendum 3. Each item is examined and the state of the Working Draft is categorized for that item. All MISSING and INCOMPLETE conditions must be corrected before DP registration. The text and font facilities need significant work, and we have dealt with them in a separate section later in this document.

3.1 Advanced 2D drawing capability

a. curves

- Bezier curves: MISSING.
- Rational B-splines: OK.
- Parametric spline curves: OK.
- Conics, and conic arcs: OK.

b. additional line attributes

- line cap: OK.
- line join: OK.
- mitre limit control: OK.

c. composite line primitives: OK.

d. user defined

- line types: OK.
- hatch styles: OK.
- marker types: MISSING.

e. arbitrary text path: OK.

f. filling mechanisms

- additional standardized hatch styles: MISSING.
- interpolated fill: MISSING.
- geometrically specified patterns: MISSING.

g. general transformations: 3x3 (and 2x3) transformation matrices to allow for affine and projective transformations: MISSING (except for conics).

3.2 Improved text and font support

Much work needed.

3.3 Arbitrary boundaries for clipping and shielding

OK.

3.4 Additional color models beyond RGB

- a. CIE: OK (but check against latest draft of ISO 8613 Color Addendum).
- b. CMYK: OK (but check against latest draft of ISO 8613 Color Addendum).
- c. spot colour: OK (but check against latest draft of ISO 8613 Color Addendum).

3.5 Additional image storage and transfer capabilities

Coding/compression techniques: INCOMPLETE. (cleanup parameterizations, add local color precision, improve descriptions, address the issue of compression and color).

3.6 Symbols

- a. external reference to libraries of named symbols: MISSING.
- b. user-defined internal libraries: OK (adequately covered by Addendum 1).

4. Specific Technical Comments.

4.1 Bundles

It is not described anywhere how the new attributes (line attributes, text/font attributes, etc) relate to the bundle concept of CGM and other SC24 standards. They should not be bundled. Predefined bundles are used for distinguishability where exact appearance does not count, and this is contrary to the whole reason for these "precise control" attributes. Settable bundles in static CGM only provide a macro mechanism, and the utility of such a mechanism does not justify the complexity of adding these to bundle definitions.

4.2 Clipping to Arbitrary Boundaries

The Add.3 requirements identify the need for clipping to arbitrary boundaries, and also the need for arbitrarily specified geometric fill (the latter has not been defined yet). The latter can provide the same capability as the former. Clause 4 should have text comparing and contrasting the features, and should furthermore address (and proscribe) possible recursive situations that could arise with the filling feature.

4.3 Compound Clip Indicator

There should be a COMPOUND CLIP INDICATOR which controls the clipping to arbitrary boundaries.

4.4 Line Types and User-defined Line type

- a. Are the interior endpoints of a dashed line capped? Options are: yes, no, or selectable by a new option setting element.
- b. The need for 'millimeters' and 'native device units' as modes for dash unit selector of user line type needs to be justified.
- c. There should be a 'line width relative' mode for dash unit selector for line type definition.
- d. There should be a LINE STYLE CONTINUATION function, which applies to both user defined and predefined dashed lines. The 'adaptive' parameter should be removed from user defined linetype. LINE STYLE CONTINUATION should minimally have the values "continue", "restart", or "adaptive continue". This parameter might be subject to registration of values as well, as some of the target application areas of Add.3 have very specific requirements for how a dashed line is drawn with respect to dash lengths and inking of vertices.
- e. There should be an INITIAL DASH OFFSET element, which specifies how far into the line pattern to start drawing a new polyline.
- f. The text of User Linetype needs much clarification and improvement including definition of missing terms such as "N", "gap_array" and "duty cycle".
- g. Which of the new Line Type functions affect edges. Is it intended to have equivalent functions for edges?

4.5 Shielding

Multiple disjoint shield regions are needed. This could be achieved in a number of ways.

- The definition of a single region could allow for islands, much as POLYGON SET does for filled areas.
- or there could be multiple definitions of regions, each associated with an index and tied to its shield indicator by the index.

4.6 Additional line caps

A Line Cap style of 'triangular' is needed.

4.7 Conics and conic arcs

- a. Section 5.8.X states "the conic arc entity must be positioned such that each of its axes is parallel to either the Xv axis or Yv axis". Clause 4 does not have a similar restriction. If the latter is correct, is there need for the conic arc transformation matrix?
- b. The assertion has been made, but not yet demonstrated, that the conics can all be represented by relatively simple splines. If this can be demonstrated then the conic arc entity should be removed in favor of a non-normative note describing how parabolas and hyperbolas can be realized with splines.

4.8 General transformations

The need for this was stated in the NWI and requirements. It is not addressed in the Working Draft, and it is not clear exactly what was intended by the requirements statement. This needs to be studied and clarified. In particular, do Segment and Copy transforms provide whatever capability is needed?

4.9 Splines

- a. Unified Formulation. If formulations can be demonstrated which are consistent with the work of STEP, PHIGS+, and other related areas, and if these can be shown to easily subsume and support the formulations in the Working Draft, then these formulations should be used. Such claims have been made but not demonstrated.
- b. Description. Tutorial information on the various splines and curves should be included in an annex.
- c. The "curve_type" and "H" parameters of the Parametric Spline curve are meaningless for the definition of the curve and should be removed.
- d. The "equation type" flag should be removed — to get a polynomial one may put in an array of weights that sum to zero. A void array should be degenerate or illegal.
- e. Closed curves. Clause 4 indicates a RATIONAL B-SPLINE CURVE CLOSED element, which is a Filled Area element. This is missing from Clause 5.

4.10 PEL ARRAY

- a. The compression parameters should be separated out as a distinct (control or attribute) element.
- b. Line spacing and pel spacing should be real values specified as PELS/VDC.
- c. The IMAGE APERTURE serves no useful purpose that cannot be met with CLIP RECTANGLE — it should be removed.
- d. There should be a 'local colour precision' parameter for PEL ARRAY. It should also be explicit that not all combinations of compression and l.c.p are legal (e.g., l.c.p must be 1 for CCITT).
- e. The document needs more work on colour compression techniques.
- f. There is no definition of what a PEL is, and how it differs from a cell or a pixel.

4.11 TILED PEL ARRAY

It is not clear that this is not in fact a graphical primitive but a "tiling control function". Better explanation and a name like PEL ARRAY TILING CONTROLS would help.

4.12 Color Models

The terminology "Color Model" is commonly used and should replace the terminology "Colour Representation Method" of the Working Draft.

The color models HLS and HSV are missing. There seems to be no good reason to exclude them, particularly since they are in use in the API standards now.

4.13 User Defined Marker Types

These are stated in the requirements, and are missing from the Working Draft. It should be evaluated if the need for this can be satisfied by applying the segment mechanisms of Add.1

4.14 Additional Hatch Styles

These are stated in the requirements, but there are none in the Working Draft. The requirements have not been specific on what styles were anticipated. The US simply notes this omission and has no styles to suggest at this time.

4.15 Interpolated Fill

This is stated in the requirements, but is missing from the working draft. There are very many ways to do smooth shaded fill. The activity in PHIGS and current practice in the industry should be examined for guidelines for a base proposal or initial set of methods.

4.16 Transformation of Hatches

There should be a HATCH TRANSFORM DESIGNATOR element. For engineering practices hatches do not transform, but for other applications they do. There should be a control or attribute element to control this behavior.

4.17 References to External Symbols

This capability is stated in the requirements, and is missing from the Working Draft. This capability is useful in engineering practice and for "clip art" in graphics arts. The issue of referencing external resources which are not part of or defined by this or another standard is the key issue which needs to be solved in order to include this capability. Once that issue is resolved it appears promising that the existing segmentation facility of Add.1 can be used to provide a complete mechanism.

4.18 Improved Text and Font Capability

Aside from the many specific issues identified below, the US questions the level of font resource information which is being carried in the metafile. The information can be grouped into levels of detail and preciseness. These are based on the font resource subsets of ISO 9541:

- Font Description: resource name itself, family name, design class, weight, posture, etc.
- Font metric information
- Glyph metric and kerning information
- Glyph shape definition.

There should be some amount of the highest level information in the metafile, as part of the font callout procedure. There should be none of the lowest level. How much of the intermediate levels of information should be included is an open issue.

We note that the latest draft of ISO 9541 has dropped the font information subsets concept, which we considered useful information to application communities such as computer graphics metafiles.

Following are specific technical topics relating to text.

- a. Available glyphs for TEXT primitives. The repertoire of glyphs available with the current CGM character set mechanisms, and means to access glyph collections, is not adequate. A companion standard of ISO 9541 — ISO 10036 — establishes the Association for Font Information Interchange (affi) as the registrar for glyphs and glyph collections. Affi estimates that 25000 glyphs will be registered in the next 2 years. It appears that each will have a unique 4-byte identifier. Add.3 must have a way to access glyph collections that do not fit into the traditional model of "G-sets" and character set switching defined by ISO 2022. At the least a new value of CHARACTER CODING ANNOUNCER is needed. Then the metafile might use affi 4-byte identifiers and/or mapping tables to convert 4 byte ID's to 1 or 2 byte codes.
- b. Inclusion and formulation of glyph definitions. There are many technical problems with the formulation of glyph definitions. It is the US position, however, that glyph definitions do not belong in the metafile. There are two reasons:
 - Glyph shape descriptions do not belong in the metafile; they should be part of the external font resource, which may accompany the metafile but is not

a part of it.

- The work is premature with respect to work on ISO 9541/3, which is just commencing.

Note: the US believes that CGM (Add.3) technology should be used to define an encoding of glyph shape definition, as part of an external font resource, following the functional guidelines established in ISO 9541/3, when work on that part is sufficiently advanced.

- c. BOXED TEXT should be dropped and a RESTRICTED TEXT METHOD element added such that the RESTRICTED TEXT element can function in the manner described for BOXED TEXT. The new mode element should standardize at least some text fitting options such as: as-now, text simply is adjusted not to violate box; exact-fit, the CHARACTER HEIGHT is superceded by the box height and the string width is adjusted to exactly fit the box width; width-fit, as previous, but CHARACTER HEIGHT is not altered regardless of its relationship to the box; isotropic-scale, in which the text is scaled without distortion until both dimensions fit and at least one dimension fits exactly; etc. These are just some possibilities. This should be a parameter subject to registration. There may be other desirable values which give more detail about layout algorithms.
- d. Access to ISO 9541 font resources. Methods to access all information in 9541 format font resources external to the metafile are needed. Precisely what the practical implications are remains an open issue. The shape description information must be easily accessible. At least one of the technologies used to describe the shapes in the font resource should be compatible with the CGM.
- e. Font scoring. The current wording should be clarified to indicate that the score character and the exact score position are defined in the font resource for the current font.
- f. Font Terminology. Add.3 terminology should be aligned better with 9541.

Attachment 8:
Draft Addendum Text for CGM Addendum 1

CGM Addendum 1

Part 1

Draft Copy

August 1989

Page 1

Sub-clause 0.1: Add the following at the end of the sub-clause:

This picture description includes the capability for describing static pictures. Static pictures are those where elements which may lead to dynamic effects (for example those leading to regeneration) are prohibited within the picture body.

Page 1

Sub-clause 0.3: Add the following at the end of item c):

It should also not preclude further extensions to support future standards.

Page 1

Sub-clause 0.3: Add the following at the end of item d):

It should include the capability to support ISO 7942 (GKS) static picture-capture.

Page 3

Sub-clause 0.8: Add the following at the end of the first paragraph:

The CGM specifies the elements required to support ISO 7942 (GKS) static picture-capture.

Page 3

Sub-clause 0.8: Add the following at the end of the clause:

There is a very close relationship between many of the elements in ISO 8632-1987/ADD.1 and a subset of the functions in the CGI (Computer Graphics Interface - ISO DP 9636).

Page 4

Clause 1: Add the following at the end of the first paragraph:

This picture description includes the capability for describing static images.

Page 5

Clause 2: Add the following to the list of references:

ISO DP 9636 Information processing systems - Computer Graphics - Interfacing techniques for dialogues with graphical devices (CGI). Parts 1-6.

Page 6

Sub-clause 3.1.2.6: Definition of graphical elements

Insert "primitive" between "graphical" and "element".

Page 6

Clause 3: Add the following to the list of definitions and abbreviations:

anisotropic mapping: A mapping in which the scale factors applied along each axis are not equal. This is often used in reference to the mapping from VDC to distance units on the physical drawing surface. See "isotropic mapping".

boundary: The mathematical locus that defines, in abstract VDC space, the limits of a region to be filled (for fill primitives and closed figures). The visual appearance of interior style 'hollow' consists of a depiction of the boundary obtained after clipping has been taken into account. The boundary is distinct from the edge as it includes the intersection of the region with the perimeter of the effective clipping region.

character set: The set of displayable symbols mapped to individual characters in a TEXT, APPEND TEXT, or RESTRICTED TEXT string. This corresponds to the "G-set" defined in ISO 2022. A character set is independent of the font or typeface; examples of character sets: ASCII (X3.4), German, Katakana.

clipping mode: A generic term referring to one of Line Clipping, Marker Clipping or Edge Clipping Modes. An object clipping may be either 'locus', 'shape' or 'locus then shape'.

closed figure: A compound primitive that behaves as a fill primitive of more general shape. It is formed by bracketing a sequence of line or fill primitives, edge attributes, and certain controls, with the elements BEGIN FIGURE and END FIGURE.

compound primitive: A compound primitive is specified by a sequence of CGM elements, as opposed to primitives represented by a single element. Compound text and closed figures are examples of compound primitives in the CGM.

compound text: A compound primitive formed through the use of APPEND TEXT. There may be attribute changes between portions of the resulting complete text string.

device coordinates: The coordinates native to a device; device-dependent coordinates; physical device coordinates.

device viewport: A rectangular subset of the physical drawing surface into which VDC EXTENT is mapped. See "effective viewport".

edge: The rendering of the perimeter of a filled region, controlled by edge attributes. Edges are clipped after being applied to the boundary, as distinct from the rendition of the boundary obtained from interior style 'hollow'. See "boundary".

effective viewport: The actual viewport resulting from forced isotropic mapping from the VDC extent to the viewport.

foreground colour: The colour used in the rendering process in which primitives are rendered on the drawing surface, as opposed to the BACKGROUND COLOUR or AUXILIARY COLOUR. The foreground colour is set separately for each class of primitives.

global segment: A segment that is defined in the Metafile Description (see segment). It may be referenced from within any picture.

graphic object: *****CGI*****

isotropic mapping: A mapping which is invariant with respect to direction; equal scaling in all orthogonal representational dimensions. Often used to describe the mapping from VDC to distance units on the physical drawing surface (see "anisotropic mapping").

local segment: A segment that is local to the picture in which it appears.

region: In the context of closed figures or the POLYGON SET element, an area that is explicitly or implicitly closed, that is a subset of the full area being filled. Regions can be nested, disjoint or overlapping. The boundaries of all regions are considered together when applying the interior test for filling a closed figure or POLYGON SET.

segment: A collection of primitives, primitive attributes and some additional attributes associated with the segment as a whole. See "segment attributes".

segment attribute: An attribute associated with a segment as a whole as opposed to attributes of individual primitives.

size specification mode: A generic term for Line Width Specification Mode, Edge Width Specification Mode, or Marker Size Specification Mode. A size specification mode may be VDC or scaled, the latter being referenced to a nominal size in device coordinate space.

skewed: Used to describe stroke precision text when the CHARACTER ORIENTATION vectors are non-perpendicular, CELL ARRAYS when the three defining points form a parallelogram which is not a rectangle, or a segment transformation that causes rectangles to become non-rectangular parallelograms.

Page 9

Sub-clause 4.1: Add the following at the end of the list of classes of elements:

- Segment Elements, which enable the grouping and manipulation of elements.

Page 9

Sub-clause 4.1: Add the following after the third paragraph:

Graphical output primitives and attributes may be grouped in segments. Segment attribute elements control the appearance of segments.

Page 10

Sub-clause 4.2: Add the following at the end of the sub-clause:

Groups of elements, called segments, are delimited by BEGIN SEGMENT and END SEGMENT. Each segment is uniquely identified by a segment identifier. Segments may be defined in the Metafile Descriptor or within picture bodies.

Page 10

Sub-clause 4.3: Add the following to the list after the first paragraph:

NAME PRECISION
MAXIMUM VDC EXTENT
SEGMENT PRIORITY EXTENT

NOTE: Other elements, as defined in this standard, may appear within the Metafile Descriptor within the definition of a global segment.

Page 10

Sub-clause 4.3: Add the following at the end of the sub-clause:

NOTE: It is recommended that the following elements: METAFILE VERSION, METAFILE ELEMENT LIST and METAFILE DESCRIPTION appear first in the Metafile Descriptor and in the order listed.

Page 10

Sub-clause 4.3.2: Add the following at the end of the sub-clause:

Further shorthands are defined for groups of CGM elements.

Page 11

Sub-clause 4.3.2: Add the following clauses at the end of the sub-clause:

4.3.2.3 Ver.2-static-all set

The Ver.2-static-all set may be used to indicate all the elements in the drawing-plus-control set and all the additional elements defined in ISO 8632/1-1987/Add.1.

4.3.2.4 Extended-primitives set

The Extended-primitives set may be used to indicate those primitives which are not defined in ISO 7942 (GKS). These elements are:

DISJOINT POLYLINE
RESTRICTED TEXT
APPEND TEXT
POLYGON SE.
RECTANGLE
CIRCLE
CIRCULAR ARC 3 POINT
CIRCULAR ARC 3 POINT CLOSE
CIRCULAR ARC CENTRE
CIRCULAR ARC CENTRE CLOSE
CIRCULAR ARC CENTRE REVERSED
ELLIPSE
ELLIPTICAL ARC
ELLIPTICAL ARC CLOSE
CONNECTING EDGE

4.3.2.5 Ver.2-static-gksm set

The Ver.2-static-gksm set includes elements for ISO 7942 (GKS) static picture capture. The elements included in the Ver.2-static-gksm set are:

BEGIN METAFILE
BEGIN PICTURE
BEGIN PICTURE BODY
END PICTURE
BEGIN SEGMENT
END SEGMENT
END METAFILE
METAFILE VERSION
METAFILE DESCRIPTION
VDC TYPE
INTEGER PRECISION
REAL PRECISION
INDEX PRECISION
COLOUR PRECISION
COLOUR INDEX PRECISION
NAME PRECISION
MAXIMUM COLOUR INDEX
COLOUR VALUE EXTENT
METAFILE ELEMENT LIST
METAFILE DEFAULTS REPL.
FONT LIST
CHARACTER SET LIST
CHAR CODING ANNOUNCER
MAXIMUM VDC EXTENT
SEGMENT PRIORITY EXTENT
VDC EXTENT
DEVICE VIEWPORT
DEVICE VIEWPORT MAPPING
DEVICE VPORT SPEC MODE
LINE REPRESENTATION
MARKER REPRESENTATION
TEXT REPRESENTATION
FILL REPRESENTATION
VDC INTEGER PRECISION
VDC REAL PRECISION
CLIP RECTANGLE
POLYLINE
POLYMARKER
TEXT

POLYGON
 CELL ARRAY
 GDP
 LINE BUNDLE INDEX
 LINE TYPE
 LINE WIDTH
 LINE COLOUR
 MARKER BUNDLE INDEX
 MARKER TYPE
 MARKER SIZE
 MARKER COLOUR
 TEXT BUNDLE INDEX
 TEXT FONT INDEX
 TEXT PRECISION
 CHAR EXPANSION FACTOR
 CHARACTER SPACING
 TEXT COLOUR
 CHARACTER HEIGHT
 CHARACTER ORIENTATION
 TEXT PATH
 TEXT ALIGNMENT
 CHARACTER SET INDEX
 ALT CHAR SET INDEX
 FILL BUNDLE INDEX
 INTERIOR STYLE
 FILL COLOUR
 HATCH INDEX
 PATTERN INDEX
 FILL REFERENCE POINT
 PATTERN TABLE
 PATTERN SIZE
 COLOUR TABLE
 ASPECT SOURCE FLAGS
 PICK IDENTIFIER
 ESCAPE
 MESSAGE
 APPLICATION DATA
 SEGMENT TRANSFORMATION
 SEGMENT HIGHLIGHTING
 SEGMENT DISPLAY PRIORITY
 SEGMENT PICK PRIORITY

Page 12

Add the following text to the end of sub-clause 4.4.4

MAXIMUM VDC EXTENT defines an extent which bounds the VDC extent values which may be found in the metafile. It may be, but need not be, a closest bound in the sense that it exactly equals the union of the extent rectangles in the metafile.

Page 14

Add the following sub-clause after sub-clause 4.4.6

4.4.7 Device viewport control

The device viewport specifies the region of the actual device drawing surface into which the VDC extent is to be mapped on interpretation. VDC-to-Device mapping is determined by the VDC extent, device viewport, and device viewport mapping. This type of transformation is restricted to allow only translation and scaling. No rotation or skew is possible.

The position of the device viewport is specified in one of three unit systems selected by **DEVICE VIEWPORT SPECIFICATION MODE** element:

- by fraction [0.0 - 1.0] of the available drawing surface, which allows reasonable placement and relative sizing of the viewport;
- in millimetres times a scale factor, which allows absolute sizing of images;
- in physical device units.

The DEVICE VIEWPORT MAPPING element may be used to force isotropic mapping even if the specified VDC extent and device viewport would not otherwise have led to one. In such a case, the VDC extent is mapped onto a subset of the specified device viewport on interpretation. This subset is defined by shrinking either the vertical or horizontal dimension of the specified viewport as needed to reach the required aspect ratio. This smaller "effective viewport" is then used to define the coordinate mapping from VDC to the device's coordinates. The placement of the effective viewport rectangle within the original one can be specified. This placement can be one of left, right or centred when the shrinking is horizontal, and top, bottom or centred when it is vertical. 'Left' and 'bottom' are interpreted as being towards the 'first corner' of the specified DEVICE VIEWPORT regardless of any mirroring or rotation of the viewport on the physical device. These meanings are relative to the rectangle defined by the non-inverted viewport.

For VDC Extent the coordinates can increase or decrease from the first to the second corners. If decreasing coordinates are chosen, this will lead to mirroring or rotation of primitives.

The behaviour of primitives and geometric attributes under transformations is further described in sub-clause 4.6.

If both device viewport and scaling mode appear in the same metafile then the last specified is used. If neither appear then the default values for device viewport take precedence.

4.4.8 Representations

The elements LINE REPRESENTATION, MARKER REPRESENTATION, TEXT REPRESENTATION, FILL REPRESENTATION and EDGE REPRESENTATION are used to set all of the attribute values in a bundle table entry at the same time. The attributes, which may be bundled, are described in 4.7.

Page 14

Add the following at the end of sub-clause 4.5:

Some of the control elements may appear in the Picture Descriptor if this is permitted by the formal grammar for the metafile version.

Page 15

Add the following text to the end of sub-clause 4.5.2:

There are three different clipping modes for lines, markers and edges. The required clipping mode is recorded in the metafile with the elements: LINE CLIPPING MODE, EDGE CLIPPING MODE, and MARKER CLIPPING MODE. When the CLIP INDICATOR associated with a graphical primitive is 'on', only those parts of a graphical primitive that are considered inside the effective clipping region are rendered on interpretation. The object clipping modes allow precise specification as to how clipping is applied to primitives on interpretation.

Clipping may be either 'locus', 'shape' or 'locus then shape'. Conceptually, a locus is a mathematical object like a point or line segment, while a shape is an area in 2-dimensional space. Loci are 0-, 1- or 2-dimensional subsets of real-valued 2-space. For markers and text they are points. For lines they are the individual line segments or portions of arcs. The locus of an area is the shape and the boundary. Shapes reflect the realization of geometric attributes and are generally 2-dimensional subsets of real-valued space.

'Locus' clipping is applied for each portion of a graphic object based on its mathematical location and is independent of the area it will occupy after rendering. For example, no portion of a line segment is rendered if the ideal mathematical line lies outside the effective clipping region, (even if its line width would carry some portion of the rendering of it into the clipping rectangle); no portion of a marker is rendered if its location lies outside the clipping rectangle.

If 'Locus' clipping is used, the rendering is applied to the locus of the graphic object. The resulting rendered shape areas may therefore extend outside the effective clipping region.

'Shape' clipping is applied after the abstract rendering of shape in device coordinate space. the 2-dimensional point set associated with the graphic object is intersected with the effective clipping region, which has been transformed in device coordinate space.

'Locus then shape' clipping allows the specification that both locus and shape clipping be applied to graphic objects as described above. In this case however, the rendered shape will not extend outside the effective clipping region. A thick line whose locus is outside the clip window will not have any portion visible even if its line width would carry some portion of the rendering outside the clip rectangle.

Figure ***** shows some examples of the effect of the clipping modes.

When a width or size specification mode is 'scaled', the rendering of shape proceeds in DC space after application of the VDC-to-Device mapping.

Fill and text primitives do not have associated object clipping modes, (though the edge of a fill primitive and the boundary edges of a closed figure do). Clipping for fill primitives is always consistent with shape clipping (see sub-clause 4.6.4.5). For text primitives, the type of clipping is determined by the associated text precision:

For 'string' precision text, clipping proceeds, on a per string basis, in a manner consistent with locus clipping.

For 'character' precision text, clipping proceeds, on a per character basis, in a manner consistent with locus clipping.

For 'stroke' precision text, the clipping always proceeds in a manner consistent with shape clipping.

Note that shape clipping for all text precisions is always allowed by this Standard.

Clip rectangles applied to graphical primitive elements within segments may be subject to transformations in VDC space. Intersection of clip rectangles (untransformed or transformed) may lead to resulting polygonal clipping boundaries (see 4.12.5).

Page 15

Add the following to the list of graphical primitive elements and to the list of line elements in sub-clause 4.6.1:

CIRCULAR ARC CENTRE REVERSED
CONNECTING EDGE

Page 16

Add the following before sub-clause 4.6.1:

In addition to the graphical primitive elements listed above, this Standard defines elements providing the definition of 'compound primitives' from several of the other graphical primitives. The following classes of compound primitives are defined: 'compound text' and 'closed figures'. The elements that may be used to specify compound primitives are listed in Table *****.

Table ***** Compound Primitives

Object Class	First Element	Primitives Included	Other Elements	Final Element
Compound Text	TEXT RESTRICTED TEXT (Note 1)	APPEND TEXT (Note 2) GDP (Note 5)		APPEND TEXT (Note 3) GDP (Note 5)
Closed Figure	BEGIN FIGURE	Line Primitives Fill Primitives (Note 4) GDP (Note 5)	NEW REGION	END FIGURE

Notes:

- 1 The final/not final flag is 'not final'; the primitive defines the reference point of the entire compound text primitive; the text of the primitive is entered in the buffer.
- 2 The final/not final flag is 'not final'.
- 3 The final/not final flag is 'not final'; the text of the primitive is entered in the buffer before the compound primitive is closed.
- 4 All primitives of the identified classes may be included.
- 5 Whether a GDP may contribute to compound text or closed figures, and whether or how it specifies that the compound text state or closed figure state be opened, maintained or closed, is specified with the definition of the GDP in the International Register of Graphical Items.

Graphical primitive elements and compound primitive elements may be subject to transformation in VDC space (segment and copy transformation, 4.12.4.2 and 4.12.5). Such a transformation may change the shape of some primitives. If there is a skew, a primitive initially specified as a rectangle may become a parallelogram. If there is an anisotropic scaling, a primitive initially specified as a circle may become an ellipse. Note that the shape of markers is exempt from such transformations.

Page 16

Sub-clause 4.6.1.1 add the following text to the paragraph describing CIRCULAR ARC xxx

A reverse direction arc can also be specified, this is described in 5.6.20

Page 16

Add the following at the end of sub-clause 4.6.1.1

CONNECTING EDGE A line segment connecting the last point of the preceding line element to the next point is generated during the construction of a closed figure. The next point is either the first point of the next line element or the current closure point.

Page 16

Add the following at the end of sub-clause 4.6.1.3

In version two metafiles, line clipping is controlled by the LINE CLIPPING MODE element, which can have one of the following values: 'locus', 'shape', or 'locus then shape'. However, clipping only applies if the CLIP INDICATOR is 'on'.

For 'locus' clipping, the mathematical locus of the line is clipped at the intersection with the clip rectangle before shape rendering is applied. Hence, part of the shape of a line may appear outside the clip rectangle.

For 'shape' clipping, the shape of the rendered line is clipped to the intersection with the clip rectangle, that is nothing is drawn outside the clip rectangle.

For 'locus then shape' clipping, the mathematical locus of the line is clipped, as with locus clipping, and then subsequently the rendered shape of the clipped locus is again clipped. Note that, since the mathematical locus of the line may have changes as a result of locus clipping, subsequent shape rendering and clipping may produce a different appearance of a line from either of the other two clipping modes.

If the line width is measured in VDC units it is subject to VDC-to-Device mapping (4.4.7) as well as to both segment and copy transformation (4.12.4.5 and 4.12.5). Note that the entire locus of an arc is subject to these transformations. In case of an anisotropic mapping or transformation the rendered width of the line will change with the direction of the line segment. If the line width is specified as a scale factor it is not affected by any transformations.

Page 17

Sub-clause 4.6.2.3: Add the following at the beginning of the sub-clause:

The following discussion applies to version 1 metafiles.

Page 17

Sub-clause 4.6.2.3: at the end of the first paragraph change 'is not standardized.' to the following:

is not standardized for version 1 metafiles.

Page 17

Sub-clause 4.6.2.3: Add the following at the end of the sub-clause:

In version 2 metafiles, marker clipping is controlled by the **MARKER CLIPPING MODE** element, which can have one of the following values: 'locus', 'shape' or 'locus then shape'. However, clipping applies only if the **CLIP INDICATOR** is 'on'.

For 'locus' clipping, the mathematical locus of the markers (that is the specifying points), are clipped at the intersection with the clip rectangle before shape rendering is applied. Hence, part of the shape of a marker may appear outside the clip rectangle. However the marker is visible if, and only if, its specifying point is within the clip rectangle.

For 'shape' clipping, the shape of the rendered marker symbols are clipped to the intersection with the clip rectangle, that is nothing is drawn outside the clip rectangle. Portions of the marker symbol may appear inside the clip rectangle even though the marker's locus is outside.

For 'locus then shape' clipping, the mathematical locus of the markers are clipped, as with locus clipping, and then subsequently the rendered shape of the markers are again clipped.

If the marker size is measured in VDC units, it is subject to VDC-to-Device mapping (4.4.7) as well as to both segment and copy transformation (4.12.4.5 and 4.12.5). The shape of markers is never affected by transformations, for example a circle used as a marker type shall always appear as a circle. Only the marker size may be transformed. For this purpose, conceptually, vectors with the length marker size and arbitrary orientations are transformed; the resulting marker size is determined by the orientation of the vector which maximizes the length under the transformation (euclidean norm of 2x2 transformation matrix).

If the marker size is specified as a scale factor it is not affected by any transformations.

Page 18

Add the following at the end of sub-clause 4.6.3.3

Clipping of text strings is described in 4.7.6.

The vectors specified by the **CHARACTER ORIENTATION** element (4.7.6) are subject to the VDC-to-Device mapping as well as to both segment and copy transformation.

Page 19

Add the following at the end of sub-clause 4.6.4.5

Edge clipping is controlled by the **EDGE CLIPPING MODE** element, which has the same enumerations as **LINE CLIPPING MODE**. Edges are clipped in the same way that lines are clipped, see 4.6.1.3

Page 19

Add the following sub-clause after sub-clause 4.6.4.5:

4.6.4.6 Transformation

The entire locus of rectangles, circular and elliptical filled-area elements is subject to VDC-to-Device mapping (4.4.7), segment and copy transformations (4.12.4.5 and 4.12.5). These elements may not therefore retain their specified geometry after transformation.

The vectors of the **PATTERN SIZE** element are subject to all transformations.

The edge widths are treated in exactly the same way as line widths (4.6.1.3).

Add the following sub-clause after sub-clause 4.6.7

*****to be changed in line with CGI - this is the Kona CGI text *****

4.6.8 Closed figures

3.9.5.1 Construction of closed figures. A closed figure is a fill type compound graphic object which the client constructs on the device side of the interface by invoking BEGIN FIGURE, an ordered sequence of line and fill primitives (and optionally attributes and NEW REGION functions), and END FIGURE. Edge attribute values are associated locally with the edge portions, and fill attribute values associated globally with the complete graphic object; in addition, certain general attribute values are associated locally with edge portions and globally for the interior of the fill object. The entire graphic object then travels through the CGI pipeline, and is rendered as a single unit with the parity fill algorithm described elsewhere.

3.9.5.1.1 Closure Point. The first point of the first line primitive in a new region is the closure point for that region. The Virtual Device retains this closure point for use in closing the region. When the region is closed (with a NEW REGION or END FIGURE function, or by invoking a fill primitive which begins a new region), a line segment from the last point of the last line primitive in the region to this closure point is added by the Virtual Device to the figure, unless these points are already coincident.

3.9.5.1.2 Regions. The closed figure consists of one or more regions; a region has a closed boundary which may be concave, convex, and self crossing. A region is formed either by invoking a fill type primitive in FIGURE OPEN state (which closes the last region and contributes one or more complete regions), by invoking NEW REGION to start new regions to be formed from line primitives, or by a final invocation of END FIGURE. A closed figure constructed from only line primitives without use of NEW REGION consists of a single region.

The NEW REGION function may appear anywhere in the closed figure. If the current region is closed, the function is ignored. If the current region is open, an implicit boundary portion is added from the last point of the last primitive to the current closure point unless CONNECTING EDGE has been invoked after the last line primitive, in which case an explicit boundary portion and edge portion is added instead.

3.9.5.2 Boundaries and Edges. Each region consists of a combination of implicit boundary portions and edge portions.

3.9.5.2.1 Explicit. Explicit boundary portions and edge portions are those added by client invocation of primitives in state FIGURE OPEN. These are generated in the following situations:

- for fill primitives other than POLYGON SET, the complete edge becomes an explicit boundary portion and edge portion in the closed figure.
- for line primitives, those portions which would be rendered outside of state FIGURE OPEN become explicit boundary portions and edge portions.

- For DISJOINT POLYLINE, in particular, only the segments from the first point to the second point, from the third point to the fourth point, and so on, become explicit boundary portions and edge portions when incorporated into closed figures.

- A CONNECTING EDGE primitive which precedes an action which would normally have added an implicit boundary portion to the figure either to close a region (including closing the figure itself) or to connect two line primitives results in the portion added being an explicit boundary portion and edge portion. CONNECTING EDGE preceding or following DISJOINT POLYLINE or POLYGON SET does not affect the interpretation of those functions with respect to boundaries and edges.

Edge portions take associated edge attribute values from the state list; as these state list entries can be changed between the primitives that result in edge portions in FIGURE OPEN state, each edge portion has a distinct set of attribute values associated with it.

3.9.5.2.2 Implicit. Edge attributes are never associated with implicit boundary portions. Implicit boundary portions are only rendered for interior style hollow and are a special representation of the interior, not a representation of any portion of the edge.

Implicit boundary portions are added by the CGI device to the figure definition under the following circumstances:

- invocation of NEW REGION, END FIGURE, or a fill primitive when the current region has not been explicitly closed and CONNECTING EDGE has not been invoked since the last line primitive: an implicit boundary portion is added from the last point of the last primitive to the current closure point to close the region.

- when the last point of the preceding line primitive is not coincident with the first point of the current line primitive, an implicit boundary portion is created to connect the last point of the preceding line primitive to the first point of the current line primitive.

- the portions of a DISJOINT POLYLINE which would not normally be rendered (i.e., from the second point to the third point, from the fourth point to the fifth point, and so on) result in implicit boundary portions. (These are additional to the ones which may be added to connect to a preceding or following line primitive or to effect region closure after the DISJOINT POLYLINE.)

- the portions of polygon set as described below.

3.9.5.2.3 Conditions under which no boundary or edge is added. No boundary or edge portion is ever created connecting two regions, regardless of how those regions were created or closed.

3.9.5.3 Contribution of Primitive Functions to the Figure.

3.9.5.3.1 Contribution of Line Functions to the Figure [continue with what was 3.9.5.1.1 Add this before final paragraph of the section:]

CONNECTING EDGE

If the region is open, the start point of the connecting edge is the last point of the last line primitive, and the end point of the connecting edge is either the first point of the following primitive or the current closure point as described above. If the connecting edge would be of zero length (i.e., if the two points it connects are coincident), the function is ignored. As

with other line primitives, the edge attribute values in effect at the time it is invoked are associated with any edge portion generated by this function.

If the current region is not open, invocations of the CONNECTING EDGE function are ignored (i.e., CONNECTING EDGE cannot be used to connect regions).

CONNECTING EDGE is a primitive, not a modal setting: it must be invoked once for each connecting edge desired in the figure, and once used, no longer applies to subsequent opportunities for explicit connection.

Invoking CONNECTING EDGE multiple times after a line primitive results in the last instance (with its associated attributes) being used.

[continue with last paragraph of section]

3.9.5.3.2 Contribution of Fill Functions to the Figure. [replaces 3.9.5.1.2] Each fill primitive contributes a complete region to the figure (POLYGON SET may contribute more than one), after first closing the current region if it is open. The CGI device performs an implicit NEW REGION before and after a fill primitive invoked in FIGURE OPEN state (i.e., a fill primitive leaves the current region closed, and the next primitive begins a new region).

The unclipped boundary of each fill primitive contributes to the unclipped boundary of the closed figure; the locus of its interior does not affect the boundary definition.

Contribution of POLYGON SET to figure construction:

- A POLYGON SET is considered to contribute one or more complete regions. If the current region has not been closed, an implicit NEW REGION is performed before the POLYGON SET is added to the figure definition. If the POLYGON SET does not end with a point whose edge-out flag is "close visible" or "close invisible", an implicit NEW REGION is performed after the POLYGON SET.
- Sequences of points with edge-out flag "visible" are treated as if they were polylines, terminating with the first point with a different edge-out flag. Each such polyline becomes an edge portion of the boundary of the figure. The edge attribute values in effect when POLYGON SET is invoked are associated with any edge portion added in this way.
- Sequences of points with edge-out flag "invisible" contribute implicit boundary portions which are polylines joining the points in the sequence, but not edges. Edge attribute values are not associated with these.
- Points with edge-out flag "close invisible" generate the equivalent of a NEW REGION, generating an implicit boundary portion from this point to the current closure point if these are not coincident, and closing the current region.
- Points with edge-out flag "close visible" generate the equivalent of a CONNECTING EDGE followed by a NEW REGION, resulting in an edge portion from this point to the current closure point if these are not coincident. The edge attribute values in effect when POLYGON SET is invoked are associated with any edge portion added in this way.
- POLYGON SET does not affect the value of the edge visibility value in the state list.

3.9.5.3.3 GDP. A GDP which is defined as a line type primitive must specify which is the first point and the last point in its point list, with respect to closed figure construction.

Such GDP's are assumed to contribute to a closed figure a boundary corresponding to the unclipped locus which would be rendered if the function were invoked when not in FIGURE OPEN state; any other behaviour shall be documented explicitly in the GDP description. A GDP which is defined as being a fill type primitive function is treated as in the previous section; any variation or special handling in state FIGURE OPEN shall be explicitly documented in the GDP description.

3.9.5.4 [replace 3.9.5.1.4-6] Association of attributes.

3.9.5.4.1 Local attributes are those associated with each edge portion, which can vary from edge portion to edge portion within the compound object. The local attributes for closed figures are the set of edge attributes.

3.9.5.4.2 Global attributes are those associated with the compound object as a whole rather than its component parts. The functions which set their state list values may be invoked during FIGURE OPEN state, and have their usual effect on the corresponding state list entries. The values associated with the closed figure are those in the state list when END FIGURE is invoked to complete the object formation (even in the event of figure buffer overflow during construction). Global attributes of closed figures are the set of fill attributes, CLIP INDICATOR, CLIP RECTANGLE, EDGE CLIPPING MODE, and PICK ID. In particular, note that Clip Rectangle and Clip Indicator are associated with the compound object, but not applied during graphic object formation (the graphic object is formed from the unclipped locus of each primitive function invoked in FIGURE OPEN state).

3.9.5.4.3 There is a set of attributes which are local attributes with respect to the edge portions, but which are associated globally with the interior. This set consists of AUXILIARY COLOUR (and its corresponding colour selection mode in which set), TRANSPARENCY, and DRAWING MODE. In order to use a different value for the interior from that for any of the edge portions, the appropriate attribute function should be invoked just prior to invoking END FIGURE with the value to be used for the interior.

*****end CGI text *****
 *****all examples need redoing - liaison with CGI needed *****

4.6.8.8. Examples

Note: The clear text encoding is used here for illustration purpose.

*****need updating *****

Example 1.

This example uses the Arc Centre command to create a doughnut shape. The following commands are used:

```
BeginFig;  
ArcCtr 50,50 1,0 1,0 45;  
ArcCtr 50,50 1,0 1,0 40;  
EndFig;
```

Note that this figure can also be obtained by the sequence:

```
Begin Fig;  
Circle 50,50 45;  
Circle 50,50 40;  
EndFig;
```

Example 2

This example uses the Elliptical Arc command to create a box with rounded corners. The following commands are used:

```
BeginFig:
% All straight edges connecting the elliptical arcs %
% are drawn as implicit edges %
EllipArc 75.82 90.82 75.110 1.0 0.1;
EllipArc 25.82 25.110 10.82 0.1 -1.0;
EllipArc 25.38 10.38 25.10 -1.0 0.-1;
EllipArc 75.38 75.10 90.38 0.-1 1.0;
EndFig:
```

Example 3.

This example uses the Elliptical Arc command, showing how CDP order can be used to change the sweep direction. The lines indicate the short angles between the CDP's. The following commands are used:

```
BeginFig:
% All straight edges connecting the elliptical arcs %
% are drawn as implicit edges %
% The first arc is swept in a counterclockwise direction %
EllipArc 60.50 60.100 -10.50 0.1 0.-1;
% The second arc is swept in a clockwise direction %
EllipArc 60.50 60.10 0.50 0.-1 0.1;
```

Endfig:

Page 39

Add the following sub-clauses after sub-clause 4.7.8

4.7.9 Pick Identifier

The Pick Identifier is associated with graphical primitive elements within segments (see clause 4.12). It is the only attribute element which does not affect the appearance of a graphical primitive element. It merely establishes a means of identification of primitives within segments at metafile interpretation. PICK IDENTIFIER has no graphical effect and is available for application dependent communication between interpreters and generators.

4.7.10 Global and local attributes and controls

For the purpose of compound primitive definition (see 4.6) a further classification of attributes and control elements into 'global' and 'local' attribute and control elements is introduced. Global elements apply to compound primitives as a whole, while local elements apply separately to the component graphical primitives of a compound primitive.

Page 40

Add the following sub-clauses after sub-clause 4.11:

4.12 Segment elements

4.12.1 Introduction

In the CGM, graphical primitive elements or compound primitives, attribute setting elements and certain control elements may be grouped in segments as well as being invoked outside segments. They may also be defined as global segments, within the Metafile Descriptor, and can then be copied into a picture. Each segment is identified by a unique segment identifier. Segments may have the attributes:

- a. transformation;
- b. highlighting;
- c. display and pick priority;

These may be defined at segment definition time, before the first primitives of the segment, and shall not be changed thereafter for static picture-capture metafiles.

Only elements stored inside segments are affected by the segment attributes.

The segment elements are:

COPY SEGMENT
INHERITANCE FILTER
CLIP INHERITANCE
SEGMENT TRANSFORMATION
SEGMENT HIGHLIGHTING
SEGMENT DISPLAY PRIORITY
SEGMENT PICK PRIORITY

BEGIN SEGMENT and END SEGMENT are delimiter elements rather than segment elements.

4.12.2 Global and local segments

There are two types of segments: local segments and global segments. Both contain primitives and attributes which can be manipulated in the manner described above. Local Segments have no existence beyond the bounds of the picture body in which they are defined. Defining a local segment in a picture automatically includes that segment in the picture's image. In contrast, global segments can be referenced by any of the pictures in the metafile in which they are defined.

4.12.2.1 Location of and access to global segments.

A global segment is delimited by the BEGIN SEGMENT and END SEGMENT elements. Global segments are defined in the Metafile Descriptor. They are not a part of any picture within the metafile. They must be accessed from within individual pictures by the COPY SEGMENT (4.12.5) element. The COPY SEGMENT element incorporates the segment into the open picture in the same way for both local and global segments.

4.12.2.2 Allowable elements in MD and GSS states

BEGIN SEGMENT is the only segment-related element that is allowed within the Metafile Descriptor State (MDS) (see Table 3(a), the Metafile State Table). BEGIN SEGMENT changes the state to Global Segment State (GSS).

4.12.2.3 References to global segments

Within pictures, no elements are allowed that would modify the contents or default appearance of global segments (see Table 3(a)). This restriction preserves the logical independence of pictures and the ability to randomly access pictures. The only allowable references to global segments within pictures are by using the COPY SEGMENT element.

4.12.2.4 Association of control and attribute elements and attribute elements with primitives inside segments

The current modal values of control and attribute elements are associated with the primitives inside local segments. The modal values established by setting control or attribute elements within a segment remain outside the segment until they are explicitly changed.

Control and attribute elements are bound in global segments as they are in local segments. Upon the occurrence of BEGIN METAFILE, every element that is modally defined and bound to primitives (Metafile Descriptor elements defining modes and precisions, Picture Descriptor elements, Control elements, Attribute elements and Segment Control elements) has a default value. Conceptually the set of all of these define a "Modal State List".

The Metafile Descriptor is processed sequentially. Throughout the Metafile Descriptor, modal MD elements modify the MD entries in the state list and occurrences (possibly multiple) of the METAFILE DEFAULTS REPLACEMENT element allow manipulation (outside of GSS state) of the rest of the modal elements (as well as explicitly changing the defaults). Within GSS state the allowable modal elements (control, attribute, and segment attribute) also alter the contents of the Modal State List. The values of modal elements that are in effect upon BEGIN PICTURE are the default values for that picture, whether they are implicit (defined in the Standard) or explicit (that is by values set in the Metafile Defaults Replacement).

4.12.3 Delimiting and naming segments

The contents of a segment are delimited by the elements BEGIN SEGMENT and END SEGMENT. The elements in between these two delimiters are a part of that segment. Each segment has an identifier associated with it. No two global segments may have the same identifier and no local segment may have an identifier which is the same as either a local segment in the same picture or the same as a global segment.

4.12.4 Segment attributes

4.12.4.1 Introduction

The segment attributes associated with each segment control its display. Segment attributes can be set only while the segment is open. These may be defined at segment definition time, only before the first primitives of the segment, and may not be changed thereafter. When a segment is opened with the BEGIN SEGMENT element, the segment's attributes are set to their default values. Segment attributes, if set, shall be set immediately after the BEGIN SEGMENT element and before any other type of element. This structure is shown below.

```
BEGIN SEGMENT (Segment identifier)
Segment attributes
Allowed primitives, attributes and control elements in any order
END SEGMENT
```

4.12.4.2 Segment transformation

The segment transformation is a coordinate transformation associated with each segment and applies to all graphical primitives in the identified segment and will be used on interpretation. Clipping rectangles are not transformed by the segment transformation. It allows scaling, translation, and rotation of segments to be defined during segment definition.

The segment transformation is a transformation of VDC space to VDC space and is distinct from the VDC-to-Device mapping which is a transformation of VDC space to device coordinates.

The transformation attribute of a segment may be defined by the SEGMENT TRANSFORMATION element during the segment definition. A segment transformation is represented by a 2x3 matrix, composed of a 2x2 scaling and rotation portion, and a 2x1 translation portion. The default segment transformation is represented by the identity transformation. If the SEGMENT TRANSFORMATION element is not stored in the metafile, then all coordinate data is mapped using only the VDC-to-Device mapping. If the SEGMENT TRANSFORMATION is stored in the metafile, it is applied before the application of the VDC-to-Device mapping.

The use of segment transformations may produce coordinates that cannot be expressed within the VDC range. This is handled in an interpretation dependent way.

4.12.4.3 Segment highlighting

Segment highlighting can take one of two values, NORMAL and HIGHLIGHTED. The setting of this attribute selects one of these two states for the segment.

4.12.4.4 Segment display priority

The display priority attribute of a segment determines how overlapping segments are displayed. Segments with higher display priorities will be displayed as if they were in front of segments with lower display priorities. The segment display priority may be normalized to the continuous range of real numbers, zero to one, by applying the minimum extent and maximum extent values provided by the Metafile Descriptor element SEGMENT PRIORITY EXTENT. Interpretation of SEGMENT PICK PRIORITY has no graphical effect. Its generation and interpretation are implementation dependent.

4.12.4.5 Segment pick priority

The pick priority attribute of a segment is used to resolve the picking of segments which overlap. The segment pick priority may be normalized to the continuous range of real numbers, zero to one, by applying the minimum extent and maximum extent values provided by the Metafile Descriptor element SEGMENT PRIORITY EXTENT.

4.12.5 Copy segment and inheritance

The COPY SEGMENT inserts the elements of the referenced segment into the picture at the point of occurrence of the element.

The elements copied may be altered in a variety of ways:

- a. The inheritance filter mechanism controls whether individual attribute values are reapplied to the elements
- b. The clip inheritance mechanism controls whether the primitives in the segment are clipped to the current clip rectangle or to a combination of the current and the segment clipping rectangles.
- c. The primitive elements are transformed by the copy segment transformation and optionally by the segment transformation of the copied segment according to the rules for transformation

COPY SEGMENT has a transformation matrix as a parameter. The copy segment transformation is applied to graphical primitives before they are copied. This also applies to clipping rectangles in the segment (see below). Graphical primitives may be transformed to alter their location, size, and orientation.

A segment may be referenced by the COPY SEGMENT element, either within a picture or in a global segment. The attributes associated on interpretation can be those bound to the segment being copied, or can be imposed by the inclusion of the INHERITANCE FILTER element.

The clipping associated with a segment can be that associated with the picture at the time of the copy or can be a combination of the current clipping and the segment clipping when the CLIP INHERITANCE element is used.

The inheritance filter mechanism allows the use of the current values of attributes and controls to be associated with the copied segment in place of the attributes and controls bound to the primitives when the segment was created. The attributes and controls to be associated with the segment can be all attributes or can be a subset of attributes. The attributes and controls are selected using the INHERITANCE FILTER element. The attributes and controls can be selected using individual or group names for attributes, controls and ASFs. The elements which can be selected are shown in Table ***** for attributes and controls and in Table ***** for ASFs.

The individual element names as well as the group names are those in the table showing attribute groups below.

If an attribute or group of attributes designated in the filter selection list is set to 'state_list', graphics objects inherit that attribute or group of attributes from the current modal values when a segment is copied.

If an attribute or group of attributes designated in the filter selection list is set to 'segment', that attribute or group of attributes is unaffected (in all graphics objects employing them) by the corresponding current state list when a segment is copied.

The default inheritance filter setting value is 'segment' for all attributes and controls.

Table ***** Inheritance Filter Selection Names for Attributes

Attribute Group Name	Individual Attribute Name
LINE ATTRIBUTES	LINE BUNDLE INDEX LINE TYPE LINE WIDTH LINE COLOUR LINE CLIPPING MODE
MARKER ATTRIBUTES	MARKER BUNDLE INDEX MARKER TYPE MARKER SIZE MARKER COLOUR MARKER CLIPPING MODE
TEXT ATTRIBUTES	TEXT BUNDLE INDEX TEXT FONT INDEX TEXT PRECISION CHARACTER EXPANSION FACTOR CHARACTER SPACING TEXT COLOUR
CHARACTER ATTRIBUTES	CHARACTER HEIGHT CHARACTER ORIENTATION TEXT PATH
FILL ATTRIBUTES	TEXT ALIGNMENT FILL BUNDLE INDEX INTERIOR STYLE FILL COLOUR HATCH INDEX PATTERN INDEX
EDGE ATTRIBUTES	EDGE BUNDLE INDEX EDGE TYPE EDGE WIDTH EDGE COLOUR EDGE VISIBILITY EDGE CLIPPING MODE
PATTERN ATTRIBUTES	FILL REFERENCE POINT PATTERN SIZE
OUTPUT CONTROL	AUXILIARY COLOUR
PICK IDENTIFIER	TRANSPARENCY
ALL ATTRIBUTES	PICK IDENTIFIER
ALL	All attributes All attributes and control elements

Table XXXX Inheritance Filter Selection Names for Aspect Source Flags

ASF Group Name	Individual ASF Name
LINE ASFS	LINE TYPE ASF LINE WIDTH ASF LINE COLOUR ASF
MARKER ASFS	MARKER TYPE ASF MARKER SIZE ASF MARKER COLOUR ASF
TEXT ASFS	TEXT FONT INDEX ASF TEXT PRECISION ASF CHARACTER EXPANSION FACTOR ASF CHARACTER SPACING ASF
FILL ASFS	TEXT COLOUR ASF INTERIOR STYLE ASF FILL COLOUR ASF HATCH INDEX ASF PATTERN INDEX ASF
EDGE ASFS	EDGE TYPE ASF EDGE WIDTH ASF EDGE COLOUR ASF
ALL ASFS	All aspect source flags

An example of the COPY SEGMENT element with the INHERITANCE FILTER element is as follows:

```

BEGIN METAFILE "...
.
BEGIN SEGMENT (1)
    LINE COLOUR (blue)
    POLYLINE                                     (blue solid line)
END SEGMENT

BEGIN DEFAULTS REPLACEMENT
    LINE TYPE (dash)
END DEFAULTS REPLACEMENT

BEGIN SEGMENT (2)
    LINE COLOUR (red)
    INHERITANCE FILTER (LINE ATTRIBUTES.STATE_LIST)
    COPY SEGMENT (1)                             (red dashed line)
    POLYLINE                                     (red dashed line)
    INHERITANCE FILTER (LINE ATTRIBUTES.SEGMENT)
    COPY SEGMENT (1)                             (blue solid line)
    POLYLINE                                     (red dashed line)
END SEGMENT

BEGIN PICTURE "...
BEGIN PICTURE BODY
LINE COLOUR (green)
INHERITANCE FILTER (LINE ATTRIBUTES.SEGMENT)
COPY SEGMENT (2)                                red dashed line
                                                red dashed line
                                                blue solid line
                                                red dashed line
                                                green dashed line

POLYLINE
INHERITANCE FILTER (LINE ATTRIBUTES.STATE_LIST)
COPY SEGMENT (2)                                green dashed line
                                                green dashed line
                                                green dashed line
                                                green dashed line

BEGIN SEGMENT (3)

```

LINE COLOUR (red)	
COPY SEGMENT (1)	red dashed line
INHERITANCE FILTER (LINE ATTRIBUTES.SEGMENT)	
COPY SEGMENT (1)	blue solid line
END SEGMENT	
LINE COLOUR (green)	
COPY SEGMENT (3)	red dashed line
	blue solid line
INHERITANCE FILTER (LINE ATTRIBUTES.STATE_LIST)	
COPY SEGMENT (3)	green dashed line
	green dashed line
END PICTURE	
END METAFILE	

Clipping is not included in the INHERITANCE FILTER. There is a separate element that controls clipping behaviour CLIP INHERITANCE. Its values may be either 'state_list' or 'intersection'.

If the value is 'state_list', then the clip rectangle associated with primitives in the copied segment is that of the last CLIP RECTANGLE encountered in the metafile element sequence prior to the COPY SEGMENT element, that is the value in the "modal state list".

If the value is 'intersection', and if both the modal state list clip indicator of the segment are 'on', then the intersection of the copied segment is the intersection of the modal state list rectangle and the primitive's associated clip rectangle. If either indicator is 'off', then there is no contribution from its associated rectangle. To illustrate: if TA is the copy transformation:

```

BEGIN SEGMENT A
CLIP RECTANGLE R1
POLYLINE P1
END SEGMENT

CLIP INHERITANCE "INTERSECTION"
CLIP RECTANGLE R2
POLYLINE P2
COPY SEGMENT (A.TA)
POLYLINE P3

```

P2 and P3 are transformed by R2. P1 is transformed by R2 (combined with) TA(R1). This may be an 8-sided convex polygon, if TA causes rotation and skewing.

This composition of clipping rectangles continues, however many levels deep the segment hierarchy is nested. For example:

```

BEGIN SEGMENT A
CLIP RECTANGLE R0
POLYLINE P0
CLIP RECTANGLE R1
POLYLINE P1
END SEGMENT

BEGIN SEGMENT B
CLIP RECTANGLE R2
CLIP INHERITANCE "INTERSECTION"
COPY SEGMENT (A.TA)
END SEGMENT

CLIP RECTANGLE R3
CLIP INHERITANCE "INTERSECTION"

```

COPY SEGMENT (B.TB)
POLYLINE P3

The effective clipping "rectangles" are:

for P1: TB(R2 intersection TA(R1)) intersection R3
for P2: TB(R2) intersection R3
for P3: R3
for P0: TB(R2 intersection TA(R0)) intersection R3

From this example it can be seen that the effective clipping "rectangle" can in fact be an arbitrary convex polygon. Annex D contains recommended fallback for interpreters which cannot perform such clipping.

Segment Transformations are never applied to clipping boundaries. The default value for CLIP INHERITANCE is 'state_list'.

4.12.6 Save and Restore Primitive Context

Two elements are provided to save and restore a context, that is attributes and control elements. This capability allows a list of attributes and control elements to be stored in the metafile which can be referenced by name at a later point in the metafile. This capability can be used to save and restore attributes and control elements in conjunction with opening and closing segments.

*****should we put more details and table in here for the save and restore?*****

Page 41

Add the following text after the state diagram

NOTE: Many elements allowed in state PO can also occur in the METAFILE DEFAULTS REPLACEMENT.

Add the following table following the state diagram

Table 3.1: CGM Elements by their allowed states

CGM Element	CGM States									
	PCS	MDS	DR (4)	GSS	PDS	POS	TOS	LSS	FOS	
BEGIN METAFILE (note 1)										
BEGIN PICTURE	X	X								
BEGIN PICTURE BODY					X					
END PICTURE						X				
BEGIN SEGMENT		X				X				
END SEGMENT				X				X		
END METAFILE	X									
METAFILE VERSION		X								
METAFILE DESCRIPTION		X								
VDC TYPE		X	X							
INTEGER PRECISION		X	X							
REAL PRECISION		X	X							
INDEX PRECISION		X	X							
COLOUR PRECISION		X	X							
COLOUR INDEX PRECISION		X	X							
NAME PRECISION		X	X							
MAXIMUM COLOUR INDEX		X	X							
COLOUR VALUE EXTENT		X	X							
METAFILE ELEMENT LIST		X								
METAFILE DEFAULTS REPL.		X								
FONT LIST		X	X							
CHARACTER SET LIST		X	X							
CHAR CODING ANNOUNCER		X	X							
METAFILE CATEGORY		X	X							
MAXIMUM VDC EXTENT		X	X							
SEGMENT PRIORITY EXTENT		X	X							
SCALING MODE			X		X					
COLOUR SELECTION MODE			X	X	X	X		X		
LINE WIDTH SPEC MODE			X	X	X	X		X		
MARKER SIZE SPEC MODE			X	X	X	X		X		
EDGE WIDTH SPEC MODE			X	X	X	X		X		
VDC EXTENT			X		X					
BACKGROUND COLOUR			X		X					
DEVICE VIEWPORT			X		X					
DEVICE VIEWPORT MAPPING			X		X					
DEVICE VPORT SPEC MODE			X		X					
LINE REPRESENTATION			X		X					
MARKER REPRESENTATION			X		X					
TEXT REPRESENTATION			X		X					
FILL REPRESENTATION			X		X					
EDGE REPRESENTATION			X		X					
VDC INTEGER PRECISION			X	X		X		X	X	
VDC REAL PRECISION			X	X		X		X	X	
AUXILIARY COLOUR			X	X		X	X	X	X	
TRANSPARENCY			X	X		X	X	X	X	

CGM	CGM States								
Element	PCS	MDS	DR	GSS	PDS	POS	TOS	LSS	FOS
CLIP RECTANGLE			X	X		X		X	
CLIP INDICATOR			X	X		X		X	
LINE CLIPPING MODE			X	X		X		X	
MARKER CLIPPING MODE			X	X		X		X	
EDGE CLIPPING MODE			X	X		X		X	
BEGIN FIGURE				X		X		X	
END FIGURE									X
NEW REGION									X
IMPLICIT EDGE VISIBILITY			X	X		X		X	X
SAVE PRIMITIVE ATTS				X		X		X	
RESTORE PRIMITIVE ATTS				X		X		X	
POLYLINE				X		X		X	X
DISJOINT POLYLINE				X		X		X	X
POLYMARKER				X		X		X	
TEXT				X		X		X	
RESTRICTED TEXT				X		X		X	
APPEND TEXT							X		
POLYGON				X		X		X	X
POLYGON SET				X		X		X	X
CELL ARRAY				X		X		X	
GDP				X		X		X	X
RECTANGLE				X		X		X	X
CIRCLE				X		X		X	X
CIRCULAR ARC 3 POINT				X		X		X	X
CIRCULAR ARC 3 PT CLOSE				X		X		X	X
CIRCULAR ARC CENTRE				X		X		X	X
CIRC ARC CENTRE CLOSE				X		X		X	X
ELLIPSE				X		X		X	X
ELLIPTICAL ARC				X		X		X	X
ELLIPTICAL ARC CLOSE				X		X		X	X
CIRC ARC CENTRE REV'D				X		X		X	X
CONNECTING EDGE									X
LINE BUNDLE INDEX			X	X		X		X	
LINE TYPE			X	X		X		X	
LINE WIDTH			X	X		X		X	
LINE COLOUR			X	X		X		X	
MARKER BUNDLE INDEX			X	X		X		X	
MARKER TYPE			X	X		X		X	
MARKER SIZE			X	X		X		X	
MARKER COLOUR			X	X		X		X	
TEXT BUNDLE INDEX			X	X		X	X	X	
TEXT FONT INDEX			X	X		X	X	X	
TEXT PRECISION			X	X		X	X(3)	X	
CHAR EXPANSION FACTOR			X	X		X	X	X	
CHARACTER SPACING			X	X		X	X	X	
TEXT COLOUR			X	X		X	X	X	
CHARACTER HEIGHT			X	X		X	X	X	
CHARACTER ORIENTATION			X	X		X		X	
TEXT PATH			X	X		X		X	
TEXT ALIGNMENT			X	X		X		X	
CHARACTER SET INDEX			X	X		X	X	X	
ALT CHAR SET INDEX			X	X		X	X	X	
FILL BUNDLE INDEX			X	X		X		X	
INTERIOR STYLE			X	X		X		X	

CGM Element	CGM States								
	PCS	MDS	DR	GSS	PDS	POS	TOS	LSS	FOS
FILL COLOUR			X	X		X		X	
HATCH INDEX			X	X		X		X	
PATTERN INDEX			X	X		X		X	
EDGE BUNDLE INDEX			X	X		X		X	X
EDGE TYPE			X	X		X		X	X
EDGE WIDTH			X	X		X		X	X
EDGE COLOUR			X	X		X		X	X
EDGE VISIBILITY			X	X		X		X	X
FILL REFERENCE POINT			X	X		X		X	
PATTERN TABLE			X		X	X			
PATTERN SIZE			X	X		X		X	
COLOUR TABLE			X		X	X			
ASPECT SOURCE FLAGS			X	X		X		X	X(2)
PICK IDENTIFIER			X	X		X		X	
ESCAPE	X	X		X	X	X	X	X	X
MESSAGE	X	X		X	X	X		X	X
APPLICATION DATA	X	X		X	X	X		X	X
COPY SEGMENT				X		X		X	
INHERITANCE FILTER			X	X		X		X	
CLIP INHERITANCE			X	X		X		X	
SEGMENT TRANSFORMATION			X	X				X	
SEGMENT HIGHLIGHTING			X	X				X	
SEGMENT DISPLAY PRIORITY			X	X				X	
SEGMENT PICK PRIORITY			X	X				X	
PIXEL ARRAY						X			

PCS Picture Closed State
 MDS Metafile Description State
 DR Defaults Replacement Mode
 GSS Global Segment State
 PDS Picture Description State
 POS Picture Open State
 TOS Text Open (Partial text) State
 LSS Local Segment State
 FOS Figure Open State

Notes:

- 1: BEGIN METAFILE is the only element allowed in the state 'Metafile Closed'
- 2: Only Edge ASFs are allowed in Figure Open State
- 3: Use of TEXT PRECISION in text open state is permitted, however the intended result is not well defined and such usage is likely to lead to unpredictable results.
- 4: Defaults replacement mode is not actually a metafile state, but is included in this table for completeness.

Page 42

Sub-clause 5.1: Add the following after the ninth paragraph which starts with the sentence: "The External Elements....":

The Segment Elements (see 5.10) provide for the grouping and manipulation of elements.

Sub-clause 5.1: Add the following at the end of the table of abbreviations of data type names:

N	Name	Identifier for segments, pick / * a set of values used with SAVE and RESTORE PRIMITIVE CONTEXT Realization is integer, range is dependent on NAME PRECISION
SN	Segment Name	Segment Identifier Realization is an integer.
VP	Viewport Point	Two VSC values representing the x and y coordinates of a point in viewport specification space
VC	Viewport Coordinate	Single real or integer value as determined by the DEVICE VIEWPORT SPECIFICATION MODE: R fraction [0..1] of default viewport I millimetres (scaled) I native device units

Add the following sub-clauses after sub-clause 5.2.5:

5.2.6 BEGIN SEGMENT

Parameters:

Segment Identifier (N)

Description:

This element demarcates the start of a segment. All subsequent elements until the next END SEGMENT will belong to this segment.

Reference:

4.2
4.12.3

5.2.7 END SEGMENT

Parameters:

None

Description:

Subsequent elements will no longer belong to a segment.

Reference:

4.2

Add the following at the end of the Description section of sub-clause 5.3.1

The CGM as defined in ISO 8632/1-1987/Add.1 is version two (2).

- Sub-clause 5.3.11: Add the following shorthand names at the end of the list given in the second paragraph of the 'Description':

VER.2-STATIC-ALL SET

EXTENDED-PRIMITIVES SET
VER.2-STATIC-GKSM SET

Page 55

Add the following sub-clauses after sub-clause 5.3.15:

5.3.16 NAME PRECISION

Parameters:

The form of the parameter depends on the specific encoding.

Description:

The precision for operands of data type name (N) is specified for subsequent data of type N. The precision is defined as the field width measured in units applicable to the specific encoding.

Reference:

4.3

5.3.17 MAXIMUM VDC EXTENT

Parameters:

first corner (P)
second corner (P)

Description:

The two corners define a rectangular extent in VDC space which bounds the values of the VDC EXTENT elements which may be found in the metafile. It may be, but need not be, a closest bound in the sense that it exactly equals the union of the extent rectangles in the metafile.

Reference:

4.4.4

5.3.18 SEGMENT PRIORITY EXTENT

Parameters:

minimum extent (I)
maximum extent (I)

Description:

The parameters represent an extent which bounds the segment display and pick priority values which will be encountered in the metafile. It need not represent the exact priorities in the metafile. The lowest display priority is zero.

References:

4.12.4.3
4.12.4.4

Page 56

Add the following note to the end of sub-clause 5.4.1 (SCALING MODE)

NOTE: If both device viewport and scaling mode appear in the same metafile then the last specified is used. If neither appear then the default values for device viewport take precedence.

Add the following sub-clauses after sub-clause 5.4.7:

5.4.8 DEVICE VIEWPORT

Parameters:

first corner (VP)
second corner (VP)

Description:

The two parameters define the opposite corners of a rectangular viewport on the device's drawing surface. These parameters are specified by the unit system selected by DEVICE VIEWPORT SPECIFICATION MODE.

The effective viewport is that area of the drawing surface onto which the VDC extent rectangle is mapped. If the current DEVICE VIEWPORT MAPPING forces isotropic mapping, and the aspect ratio is not equal to that of the device viewport, the effective viewport will be smaller than the specified viewport on one or the other axis (but not both).

If the current DEVICE VIEWPORT MAPPING does not force isotropic mapping, the effective viewport will be the same as the specified viewport. If the Device Viewport exceeds the available drawing surface, the Device Viewport is still used to determine the VDC-to-Device mapping.

Mirroring or 180 degree rotation of the image may be achieved by specifying the corners in some way other than that the first is below and to the left of the second.

NOTE: If both device viewport and scaling mode appear in the same metafile then the last specified is used. If neither appear then the default values for device viewport take precedence where these are allowed in the same category.

Reference:

4.4.7

5.4.9 DEVICE VIEWPORT SPECIFICATION MODE

Parameters:

VC specifier (one of: fraction of drawing surface
millimetres with scalefactor,
physical device units)(E)

Metric scale factor (R)

Description

This element determines how subsequent elements using the data type VC (Viewport Coordinate) or VP (Viewport Point) will be defined.

These parameters may be specified in one of three modes: fraction of drawing surface; millimetres with scale factor; or physical device units.

When the VC specifier is 'fraction of drawing surface', the value (0.0, 0.0) corresponds to the lower left corner and the value (1.0, 1.0) corresponds to the upper right corner of the default device viewport. (The default device viewport is the largest unrotated rectangular area visible on the drawing surface.) Numbers outside of the range [0.0..1.0] may be specified (see DEVICE VIEWPORT). In this case the metric scale factor is ignored.

When the VC specifier is 'millimetres with scalefactor', the metric scale factor parameter represents the distance (in millimetres) on the drawing surface corresponding to one unit in VP space. One unit in VP space represents one millimetre multiplied by the metric scale factor. The value (0,0) corresponds to the lower left corner and the values increase positively to the right and upwards.

When the VSC specifier is 'physical device units', the native units and handedness of the physical device are used. In this case the metric scale factor is ignored.

Metric scaling with a scale factor provides a device-independent means of generating output at a known scale factor. In metric mode, a scale factor of 1.0 indicates that the VC are in units of millimetres; a scale factor of 0.0254 would imply a VSC of one thousand per inch. The only allowed data type for physical device units is integer.

Reference:
4.4.7

5.4.10 DEVICE VIEWPORT MAPPING

Parameters:

Isotropy flag	(one of: not forced, forced)(E)
Horizontal alignment flag	(one of: Left, Centre, Right)(E)
Vertical alignment flag	(one of: Bottom, Centre, Top)(E).

Description:

This element determines how the coordinate mapping is derived from the VDC EXTENT and the specified DEVICE VIEWPORT. The remaining parameters are only significant if isotropy is forced by the first parameter. If so, the effective viewport is generally smaller than the specified viewport, and these parameters determine how it will be positioned within the specified viewport. 'Left' and 'bottom' are interpreted as being towards the 'first corner' of the specified DEVICE VIEWPORT, regardless of any mirroring or rotation of the viewport on the physical device.

Reference:
4.4.7

5.4.11 LINE REPRESENTATION

Parameters:

line bundle index (DX)
line type indicator (IX)
line width specifier
if line width specification mode is 'absolute',
absolute line width (VDC)
if line width specification mode is 'scaled',
line width scale factor (R)
line colour
if the colour selection mode is 'indexed',
line colour index (CI)
if the colour selection mode is 'direct',
line colour value (CD)

Description:

In the line bundle table, the given line bundle index is associated with the specific parameters.

Line type is specified and behaves as indicated in the LINE TYPE attribute element.

Line width is defined in the current LINE WIDTH SPECIFICATION MODE and is stored in the bundle table along with that mode. Thus the definition is immune to subsequent changes in the specification mode.

Line colour is defined in the current COLOUR SELECTION MODE, and is stored in the bundle table along with that mode. Thus the definition is immune to subsequent changes to the selection mode.

Which aspects are used depends on the corresponding ASFs. see the ASPECT SOURCE FLAG element.

Reference:
4.4.8

5.4.12 MARKER REPRESENTATION

Parameters:

marker bundle index (IX)
marker type indicator (IX)
marker size specifier
 if marker size specification mode is 'absolute',
 absolute marker size (VDC)

 if marker size specification mode is 'scaled',
 marker size scale factor (R)

marker colour
 if the colour selection mode is 'indexed',
 marker colour index (CI)

 if the colour selection mode is 'direct',
 marker colour value (CD)

Description:

In the marker bundle table, the given marker bundle index is associated with the specified parameters.

Marker type is specified and behaves as indicated in the MARKER TYPE attribute element.

Marker size is defined in the current MARKER SIZE SPECIFICATION MODE and is stored in the bundle table along with that mode. Thus the definition is immune to subsequent changes in the specification mode.

Marker colour is defined in the current COLOUR SELECTION MODE, and is stored in the bundle table along with that mode. Thus the definition is immune to subsequent changes to the selection mode.

Which aspects are used depends on the corresponding ASFs, see the ASPECT SOURCE FLAG element.

Reference:

4.4.8

5.4.13 TEXT REPRESENTATION

Parameters:

text bundle index (IX)
text font index (IX)
text precision (one of: string, character, stroke) (E)
character spacing (R)
character expansion factor (R)
text colour
 if the colour selection mode is 'indexed',
 text colour index (CI)

 if the colour selection mode is 'direct',
 text colour value (CD)

Description:

In the text bundle table, the given text bundle index is associated with the specified parameters.

Text font index is specified and behaves as indicated in the TEXT FONT INDEX attribute element.

Text precision is specified and behaves as indicated in the TEXT PRECISION attribute element.

Character spacing is specified and behaves as indicated in the CHARACTER SPACING attribute element.

Character expansion factor is specified and behaves as indicated in the CHARACTER EXPANSION FACTOR attribute element.

Text colour is defined in the current COLOUR SELECTION MODE, and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes to the selection mode.

Which aspects are used depends on the corresponding ASFs, see the ASPECT SOURCE FLAG element.

Reference:
4.4.8

5.4.14 FILL REPRESENTATION

Parameters:

fill area bundle index (IX)
interior style (one of: hollow, solid, pattern,hatch, empty)(E)
fill colour
 if the colour selection mode is 'indexed',
 fill colour index (CI)

 if the colour selection mode is 'direct',
 fill colour value (CD)

hatch index (IX)
pattern index (IX)

Description:

In the fill bundle table, the given fill bundle index is associated with the specified parameters.

Interior style is specified and behaves as indicated in the INTERIOR STYLE attribute element.

Fill colour is defined in the current COLOUR SELECTION MODE, and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes to the selection mode.

Hatch index indicator is specified and behaves as indicated in the HATCH INDEX attribute element.

Pattern index indicator is specified and behaves as indicated in the PATTERN INDEX attribute element.

Which aspects are used depends on the corresponding ASFs, see the ASPECT SOURCE FLAG element.

Reference:
4.4.8

5.4.15 EDGE REPRESENTATION

Parameters:

edge bundle index (IX)
edge type indicator (IX)
edge width specifier
 if edge width specification mode is 'absolute',
 absolute edge width (VDC)

 if edge width specification mode is 'scaled',
 edge width scale factor (R)

edge colour
 if the colour selection mode is 'indexed',
 edge colour index (CI)

 if the colour selection mode is 'direct',
 edge colour value (CD)

Description:

In the edge bundle table, the given edge bundle index is associated with the specified parameters.

Edge type is specified and behaves as indicated in the EDGE TYPE attribute element.

Edge width is defined in the current EDGE WIDTH SPECIFICATION MODE and is stored in the bundle table along with that mode. Thus the definition is immune to subsequent changes in the specification mode.

Edge colour is defined in the current COLOUR SELECTION MODE and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes to the selection mode.

Which aspects are used depends on the corresponding ASFs, see the ASPECT SOURCE FLAG element.

Reference:
4.4.8

Page 61

Add the following sub-clauses after sub-clause 5.5.6

5.5.7 LINE CLIPPING MODE

Parameters
mode (one of: locus, shape, locus then shape) (E)

Description
The Line Clipping Mode is set to the value specified.

Reference:
4.5.2

5.5.8 MARKER CLIPPING MODE

Parameters
mode (one of: locus, shape, locus then shape) (E)

Description
The Marker Clipping Mode is set to the value specified.

Reference:
4.5.2

5.5.9 EDGE CLIPPING MODE

Parameters
mode (one of: locus, shape, locus then shape) (E)

Description
The Edge Clipping Mode is set to the value specified.

Reference:
4.5.2

5.5.10 BEGIN FIGURE

Parameters:
none

Description:
This is the first element of a closed figure. All subsequent elements until the next END FIGURE will be part of the closed figure.

Reference:
4.6.3

5.5.11 END FIGURE

Parameters:
none

Description:
This element terminates the current closed figure.

If the current region has not yet been closed by a preceding NEW REGION or CONNECTING EDGE element and the last point of the last line element is not coincident with the current closure point, then the current subregion is closed by a line segment connecting the last point of the preceding line element to the current closure point. This line becomes a part of the boundary specification. If the region which has been previously closed is empty, or the last point of the last line element is coincident with the current closure point, then no line segment is generated by this element.

Reference:
4.6.8

5.5.12 NEW REGION

Parameters:
none

Description:
This element is used for control of subregion construction within closed figures.

If the current region has not yet been closed by a preceding NEW REGION or CONNECTING EDGE element and the last point of the last line element is not coincident with the current closure point, then the current subregion is closed by a line segment connecting the last point of the preceding line element to the current closure point. This line becomes a part of the boundary specification. If the region which has been previously closed is empty, or the last point of the last line element is coincident with the current closure point, then no line segment is generated by this element.

The first point of the next line element following a NEW REGION element becomes the new closure point, starting a new subregion.

Reference:
4.6.8

5.5.13 SAVE PRIMITIVE CONTEXT

Parameters:
Context name (N)

Description:
This element allows for the grouping and identification of the set of current values of the attribute and control elements listed in the list below as a single named entity.

Groups of elements may be saved in a picture or segment using the context name.

The attribute and control elements which may be saved by SAVE PRIMITIVE CONTEXT and restored by RESTORE PRIMITIVE CONTEXT are:

*****RE-ORDER TABLE *****

CHARACTER CODING ANNOUNCER
AUXILIARY COLOUR (Note 1)
EDGE BUNDLE INDEX
CLIP RECTANGLE (Note 3)
EDGE TYPE
CLIP INDICATOR
EDGE WIDTH (Note 2)

TRANSPARENCY
 EDGE COLOUR (Note 1)
 LINE BUNDLE INDEX
 EDGE VISIBILITY
 LINE TYPE
 EDGE CLIPPING MODE
 LINE WIDTH (Note 2)
 LINE COLOUR (Note 1)
 LINE CLIPPING MODE
 CHARACTER SET INDEX
 MARKER BUNDLE INDEX
 CHARACTER EXPANSION FACTOR
 MARKER TYPE
 CHARACTER CODING ANNOUNCER
 MARKER SIZE (Note 2)
 CHARACTER SPACING
 MARKER COLOUR (Note 1)
 CHARACTER HEIGHT
 MARKER CLIPPING MODE
 CHARACTER ORIENTATION
 FILL BUNDLE INDEX
 ALTERNATE CHARACTER SET INDEX
 FILL COLOUR (Note 1)
 TEXT BUNDLE INDEX
 TEXT FONT INDEX
 FILL REFERENCE POINT (Note 3)
 TEXT PRECISION
 INTERIOR STYLE
 TEXT COLOUR (Note 1)
 HATCH INDEX
 TEXT PATH
 TEXT ALIGNMENT
 PATTERN INDEX
 PICK IDENTIFIER
 PATTERN SIZE
 ASPECT SOURCE FLAGS

NOTES:

- 1: The COLOUR SELECTION MODE in which this value was last set is also recorded.
- 2: The corresponding specification mode in which this value was last set is also recorded.
- 3: The VDC TYPE in effect when these values are saved is also recorded

Reference:

4.12.6

5.5.14 RESTORE PRIMITIVE CONTEXT

Parameters:

Context name (N)

Description:

The attribute and control set recorded in the metafile with the last SAVE PRIMITIVE CONTEXT element are recalled on interpretation.

Reference:

4.12.6

Page 63

Add the following text to the end of the second paragraph of sub-clause 5.6.3

These instructions for the actual displayed position of a marker only apply to MARKER CLIPPING MODE focus

Page 78

Add the following sub-clause after sub-clause 5.6.19

5.6.20 CIRCULAR ARC CENTRE REVERSED

Parameters:

centrepoint (P)
DX_start, DY_start, DX_end, DY_end (4VDC)
radius (VDC)

Description:

A circular arc is drawn which is defined as follows:

DX_start and DY_start define a start vector, and DX_end and DY_end define an end vector. The tails of these vectors are placed on the centrepoint. A start ray and end ray are derived from the start and end vectors. The start and end rays are semi-infinite lines from the centrepoint in the directions of the start and end vectors respectively.

The specified radius and centrepoint define a circle. The arc is drawn in the negative angular direction (as defined by VDC EXTENT) from the intersection of the circle and the start ray (as obtained by measuring a distance 'radius' along the start ray from the centrepoint) to the intersection of the circle and the end ray.

The arc is displayed with current line element attributes.

Valid values of the vector components are those which produce vectors of non-zero length.

Valid values of 'radius' are non-negative VDC.

If the start ray and end ray are coincident, it is ambiguous whether the defined arc subtends 0 degrees or 360 degrees of central angle (see the specifications for the CIRCULAR ARC CENTRE in annex D).

Reference:

4.6

5.6.21 CONNECTING EDGE

Parameters:

none

Description:

During the construction of a closed figure a line segment connecting the last point of the preceeding line element and the next point is added to the boundary definition. The next point, which must be different from the last point, may be:

1. the first point of the next line element, or
2. the current closure point, that is in cases where CONNECTING EDGE is followed by either NEW REGION or END FIGURE.

The appearance of the connecting edge is fully determined by the edge attributes and EDGE VISIBILITY

References:

.....add.....

Add the following sub-clauses after sub-clause 5.7.35

5.7.36 PICK IDENTIFIER

Parameters:

pick identifier (N)

Description:

The pick identifier value is associated with all of the graphical primitive elements of a segment until the next PICK IDENTIFIER element. Usage of the PICK IDENTIFIER on interpretation is dependent upon the application and on the category of the metafile.

Reference:

4.7.9

Add the following sub-clause after sub-clause 5.9:

5.10 Segment elements

5.10.1 Segment control elements

5.10.1.1 COPY SEGMENT

Parameters:

segment identifier	(SN)
copy transformation matrix:	
scaling and rotation portion	(2x2xR)
translation portion	(2x1xVDC)
segment transformation application	(one of: NO, YES) (E)

Description:

The segment which is indicated by the segment identifier is referenced at this point in the metafile for copying into the picture, or into a segment when referenced from a segment, on interpretation. With the exception of the segment transformation associated with the copied segment, the identified segment is referred to as the copied segment. The segment attributes of the copied segment are ignored. Whether or not this segment is ignored is controlled by the "segment transformation application" parameter. The segment attributes of the segment in which the COPY SEGMENT may occur are unchanged by this element.

The copy transformation is applied to all primitive elements of the copied segment before they are copied into the open segment. The copy transformation is also applied to clipping rectangles under some circumstances.

The INHERITANCE FILTER element allows for control of the attribute values which are used when copying segments. This filter controls whether individual attribute values are reapplied to the graphical primitives. The effects of INHERITANCE FILTER are described in Clause 4. The way in which clipping is applied to primitives within a copied segment is controlled by CLIP INHERITANCE (see Clause 4).

The "segment transformation application" parameter controls whether or not the segment transformation associated with the copied segment will be applied as an effect of the copy process. If it is, the application of the segment transformation is never applied to a clip rectangle associated with a copied object.

Reference:

4.12.1
4.12.5

5.10.1.2 INHERITANCE FILTER

Parameters:

filter selection attribute designator (list elements or groups from:

LINE BUNDLE INDEX
 LINE TYPE
 LINE WIDTH
 LINE COLOUR
 LINE CLIPPING MODE
 MARKER BUNDLE INDEX
 MARKER TYPE
 MARKER SIZE
 MARKER COLOUR
 MARKER CLIPPING MODE
 TEXT BUNDLE INDEX
 TEXT FONT INDEX
 TEXT PRECISION
 CHARACTER EXPANSION FACTOR
 CHARACTER SPACING
 TEXT COLOUR
 CHARACTER HEIGHT
 CHARACTER ORIENTATION
 TEXT PATH
 TEXT ALIGNMENT
 FILL BUNDLE INDEX
 INTERIOR STYLE
 FILL COLOUR
 HATCH INDEX
 PATTERN INDEX
 EDGE BUNDLE INDEX
 EDGE TYPE
 EDGE WIDTH
 EDGE COLOUR
 EDGE VISIBILITY
 EDGE CLIPPING MODE
 FILL REFERENCE POINT
 PATTERN SIZE
 AUXILIARY COLOUR
 TRANSPARENCY
 LINE ATTRIBUTES
 MARKER ATTRIBUTES
 TEXT ATTRIBUTES
 CHARACTER ATTRIBUTES
 FILL ATTRIBUTES
 EDGE ATTRIBUTES
 PATTERN ATTRIBUTES
 OUTPUT CONTROL
 PICK IDENTIFIER
 ALL ATTRIBUTES
 ALL
 LINE TYPE ASF
 LINE WIDTH ASF
 LINE COLOUR ASF
 MARKER TYPE ASF
 MARKER SIZE ASF
 MARKER COLOUR ASF
 TEXT FONT INDEX ASF
 TEXT PRECISION ASF
 CHARACTER EXPANSION FACTOR ASF
 CHARACTER SPACING ASF
 TEXT COLOUR ASF
 INTERIOR STYLE ASF
 FILL COLOUR ASF
 HATCH INDEX ASF
 PATTERN INDEX ASF
 EDGE TYPE ASF
 EDGE WIDTH ASF
 EDGE COLOUR ASF

LINE ASFS
MARKER ASFS
TEXT ASFS
FILL ASFS
EDGE ASFS
ALL ASFS

Description:

The setting of the inheritance filter is modified for those attributes in the filter selection list. According to the setting, attributes are inherited from the current state lists or from the copied segment.

Reference:

4.12.5

5.10.1.3 CLIP INHERITANCE

Parameters:

clip inheritance (one of: state list, intersection)

Description:

The behaviour of clipping as applied to primitives in copied segments is defined. Simple clipping against the current rectangle in the modal state list is selected by the value 'state_list'. The value 'intersection' not only selects the clip rectangle to come from the segment but also enables an "object clipping" feature. The transformation of clip rectangles and accumulation or composition of multiple transformed rectangles is enabled, depending upon the settings of CLIP INDICATOR. See Clause 4 for a description of the mechanism.

References:

4.12.1

4.12.5

5.10.2 Segment Attribute Elements

Segment Attribute Elements, if used, shall all appear immediately after BEGIN SEGMENT, before the first element of another type. The segment identifier shall refer to the segment in which the elements are contained.

5.10.2.1 SEGMENT TRANSFORMATION

Parameters:

segment identifier (N)
transformation matrix:
 scaling and rotation portion (2x2xR)
 translation portion (2x1xVDC)

Description:

The segment transformation matrix for the identified segment is set to the specified parameter.

The default segment transformation is the identity matrix.

Reference:

4.12.4.5

5.10.2.2 SEGMENT HIGHLIGHTING

Parameters:

segment identifier (N)
highlighting (one of: normal, highlighted) (E)

Description:

The segment highlighting for the identified segment is set to the specified value. When the highlighting attribute is set to 'highlighted', the visual appearance of the segment is interpretation dependent. When the

highlighting attribute is set to 'normal', the segment is displayed according to the segment and primitive attributes.

Reference:

4.12.4.2

5.10.2.3 SEGMENT DISPLAY PRIORITY

Parameters:

segment identifier	(N)
segment display priority	(I)

Description:

The segment display priority for the identified segment is set to the specified value.

Segments with higher segment display priority appear to be in front of segments with lower segment display priorities. When the segment display priorities of two overlapping segments are the same, the order in which they appear is interpretation dependent.

Reference:

4.12.4.3

5.10.2.4 SEGMENT PICK PRIORITY

Parameters:

segment identifier	(N)
segment pick priority	(I)

Description:

The segment pick priority for the identified segment is set to the specified value. The pick priority does not affect the display of segments.

Reference:

4.12.4.4

Page 103

Clause 6: Add the following at the end of clause 6:

NAME PRECISION	encoding dependent
MAXIMUM VDC EXTENT	VDC EXTENT
SEGMENT PRIORITY EXTENT	0...255
DEVICE VIEWPORT	0..1..0..1.
DEVICE VIEWPORT SPECIFICATION MODE	fraction of drawing surface
DEVICE VIEWPORT MAPPING	forced, left, bottom
LINE REPRESENTATION	interpreter dependent
MARKER REPRESENTATION	interpreter dependent
TEXT REPRESENTATION	interpreter dependent
FILL REPRESENTATION	interpreter dependent
EDGE REPRESENTATION	interpreter dependent
LINE CLIPPING MODE	locus
MARKER CLIPPING MODE	locus

EDGE CLIPPING MODE	locus
PICK IDENTIFIER	0
INHERITANCE FILTER	segment
CLIP INHERITANCE	state list
SEGMENT TRANSFORMATION	1.0 0.1 0.0
SEGMENT HIGHLIGHTING	normal
SEGMENT DISPLAY PRIORITY	0
SEGMENT PICK PRIORITY	0

Page 104

Add the following clause after sub-clause 7.4

7.5 Conformance for Version 2 metafiles

This conformance section defines conformance for metafiles which are 'version 2'. A Computer Graphics Metafile (CGM) is said to conform to the standard if it implements precisely all the elements required for a version 2 metafile as defined in this standard. When determining conformance of a CGM, the formal grammar shall take precedence.

Page 123

Add the following to the end of sub-clause D.1:

In a static picture-capture metafile potentially dynamic effects are avoided by limiting the position of elements with such potentially dynamic effects. Thus bundle table definitions may only appear in the picture descriptor. In a metafile the effects of COLOUR TABLE and PATTERN TABLE are unspecified when they occur in a location with potentially dynamic implications. In metafiles which have a version number which is greater than one these elements may appear in the Picture Descriptor. Use of these elements in the picture body is discouraged in order to improve the portability and predictability of CGM exchange.

Page 125

Add sub-clause D.3.2.3

It is recommended that the mandatory elements in the Metafile Descriptor are written first in the descriptor and in the following order:

METAFILE VERSION
METAFILE ELEMENT LIST
METAFILE DESCRIPTOR

Page 127

Sub-clause D.4.3: replace the sentence with the following text:

DEVICE VIEWPORT, DEVICE VIEWPORT SPECIFICATION MODE, DEVICE VIEWPORT MAPPING

In the case where the VC specifier in DEVICE VIEWPORT SPECIFICATION MODE is set to either 'millimetres with scale factor' or 'physical device units' not all interpreters may be able to interpret the DEVICE VIEWPORT element as specified, and the interpretation becomes implementation dependent. Since the CGM does not specify the behaviour of an interpreter an application may wish to control the VDC-to-device mapping by mechanisms external to the CGM picture description, for example to include CGM pictures in documents.

Page 127

Add the following text to the end of the sub-clause D.4.4:

Clipping Modes

If interpreters cannot handle the 'locus' clipping mode for LINE CLIPPING MODE, MARKER CLIPPING MODE or EI CE CLIPPING MODE, then 'locus plus shape' should be used as a fallback.

Page 127

Add the following text to the end of sub-clause D.4.4

If interpreters cannot handle clipping to the parallelogram that could result from using CLIP INHERITANCE value 'intersection' the suggested fallback is to clip to the minimal circumscribing rectangle. In cases where multiple parallelograms might be composed (by intersection) to form a general convex polygon, interpreters should intersect the circumscribing rectangles to derive an effective clip rectangle.

Page 127

Add the following text to the end of the APPEND TEXT recommendations:

Changing the TEXT PRECISION in partial text state is likely to lead to unpredictable results. Generators are discouraged from doing this. Interpreters which can otherwise handle text attribute changes in partial text state should ignore this element in that state as a fallback.

Page 128

Sub-clause D.4.5: Add the following text between CIRCULAR ARC CENTRE CLOSE and Elliptical elements:

CIRCULAR ARC CENTRE REVERSED

If the start ray and end ray coincide, it is recommended that the interpreter draw the full circle.

*****clause D.4.5 - anything on closed figs *****

Page 132

Add the following sub-clause after sub-clause D.4.8

D.4.9 Segment elements

The restriction of segment attributes to be set only immediately after the BEGIN SEGMENT element and before any other element avoids any dynamic effects. If the output device cannot adjust segment priority on interpretation, segments should be displayed in order of priority.

Page 133

Sub-clause D5. Change the sentence to:

.....capabilities listed in the tables below, appropriate to the version of the metafile they want to support.

Page 133

Sub-clause D5. Change the title for Table 5 to:

Table 5(a) Suggested minimum capabilities for version 1 metafiles

Page 133

Sub-clause D.5 Add the following table after Table 5(a)

Table 5(b) Suggested additional minimum capabilities for version 2 metafiles

Capability	Minimum Suggested Interpreter Support
------------	---------------------------------------

DEVICE VIEWPORT SPECIFICATION MODE
DEVICE VIEWPORT MAPPING

LINE REPRESENTATION
MARKER REPRESENTATION
TEXT REPRESENTATION
FILL REPRESENTATION
EDGE REPRESENTATION
LINE CLIPPING MODE
MARKER CLIPPING MODE
EDGE CLIPPING MODE

fraction of drawing surface
not forced, forced
left, centre, right
bottom, centre, top
5 entries
5 entries
2 entries
5 entries
5 entries
locus, shape, locus then shape
locus, shape, locus then shape
locus, shape, locus then shape

The following annex forms a new annex F.

F Formal Grammar of the Functional Specification of the CGMADD1 Category

F.1 Introduction

This grammar is a formal definition of a standard CGM extended syntax. The encoding-independent and the encoding-dependent productions are separated, and there are subsections showing the syntax of each of the standardized encoding schemes. Details on the encoding of terminal symbols can be found in parts of this Standard that deal with the particular encoding schemes.

F.2 Notation used

<symbol>	- nonterminal
<SYMBOL>	- terminal
<symbol>*	- 0 or more occurrences
<symbol>+	- 1 or more occurrences
<symbol>?	- optional (0 or 1 occurrences)
<symbol>(n)	- exactly n occurrences, n=2,3,....
<symbol-1> ::= <symbol-2>	- symbol-1 has the syntax of symbol-2
<symbol-1> <symbol-2>	- symbol-1 or alternatively symbol-2
<symbol: meaning>	- symbol with the stated meaning
(comment)	- explanation of a symbol or a production

F.3 Detailed grammar

F.3.1 Metafile structure

<metafile>	:= <BEGIN METAFILE> <metafile identifier> <metafile descriptor> <metafile contents>* <END METAFILE>
<metafile identifier>	:= <string>
<metafile contents>	:= <extra element>* <picture> <extra element>*
<extra element>	:= <external element> <escape element>
<picture>	:= <BEGIN PICTURE> <picture identifier> <picture descriptor element>* <BEGIN PICTURE BODY> <picture content>* <END PICTURE>
<picture identifier>	:= <string>
<picture content>	:= <picture element> <segment>
<picture element>	:= <eligible control element> <graphical element> <closed figure> <primitive attribute element> <pattern table element>

	<colour table element>
	<specification element>
	<segment control element>
	<extra element>
<segment>	= <BEGIN SEGMENT>
	<segment identifier>
	<segment attribute element>*
	<eligible picture element>*
	<END SEGMENT>
<segment identifier>	::= <name>
<eligible picture element>	= <eligible control element>
	<graphical element>
	<closed figure>
	<primitive attribute element>
	<specification element>
	<segment control element>
	<extra element>

F.3.2 Metafile descriptor elements

<metafile descriptor>	= <extra element>*
	<identification>
	<optional descriptor element>*
*****change next bit to be like CGM ie no order mandated - how to do this?*****	
<identification>	= <METAFILE VERSION>
	<integer>
	<extra element>*
	<element list>
	<extra element>*
	<metafile description>+
<metafile description>	= <METAFILE DESCRIPTION>
	<string>
<element list>	= <METAFILE ELEMENT LIST>
	<element name>*
	<element name shorthand enumerated>*
<element name shorthand enumerated>	= <DRAWING>
	<DRAWING PLUS CONTROL>
	<VER.2 STATIC ALL>
	<EXTENDED PRIMITIVES>
	<VER.2 STATIC GKSM>
<optional descriptor element>	= <VDC TYPE>
	<vdc type enumerated>
	<MAXIMUM COLOUR INDEX>
	<colour index>
	<COLOUR VALUE EXTENT>
	<red green blue>(2)
	<METAFILE DEFAULTS REPLACEMENT>
	<element default>+
	
	+
	<CHARACTER SET LIST>
	<character set definition>+
	<CHARACTER CODING ANNOUNCER>

	<ul style="list-style-type: none"> <coding technique enumerated> <scalar precision> <MAXIMUM VDC EXTENT> <point> (2) <SEGMENT PRIORITY EXTENT> <minimum extent> <maximum extent> <segment> <extra element>
<vdc type enumerated>	<ul style="list-style-type: none"> = <INTEGER> <REAL>
<element default>	<ul style="list-style-type: none"> = <eligible control element> <picture descriptor element> <primitive attribute element> <segment attribute element> <extra element>
<p>***** in is8632 only escape allowed above not extra - what is right?*****</p>	
	= <string>
<character set definition>	<ul style="list-style-type: none"> = <char set enumerated> <designation sequence>
<index>	<ul style="list-style-type: none"> = <standard index value> <private index value>
<standard index value>	= <non-negative integer>
<non-negative integer>	= <integer> (greater or equal to 0)
<positive integer>	= <integer> (greater than 0)
<private index value>	= <negative integer>
<negative integer>	= <integer> (less than 0)
<positive index>	= <positive integer>
<char set enumerated>	<ul style="list-style-type: none"> = <94 CHAR> <96 CHAR> <MULTI-BYTE 94 CHAR> <MULTI-BYTE 96 CHAR> <COMPLETE CODE>
<coding technique enumerated>	<ul style="list-style-type: none"> = <BASIC 7-BIT> <BASIC 8-BIT> <EXTENDED 7-BIT> <EXTENDED 8-BIT>
<designation sequence>	= <string>
<scalar precision>	<ul style="list-style-type: none"> = <INTEGER PRECISION> <integer precision value> <REAL PRECISION> <real precision value> <INDEX PRECISION> <index precision value> <COLOUR PRECISION> <colour precision value> <COLOUR INDEX PRECISION> <colour index precision value> <NAME PRECISION> <name precision value> (these elements have encoding) (dependent parameters)

<point> ::= <vdc value> (2)

<minimum extent> ::= <integer>

<maximum extent> ::= <integer>

F.3.3 Picture descriptor elements

<picture descriptor element> => <SCALING MODE>
 <scaling spec mode enumerated>
 <metric scale factor>
| <VDC EXTENT>
 <point> (2)
| <DEVICE VIEWPORT>
 <viewport point> (2)
| <DEVICE VIEWPORT SPECIFICATION MODE>
 <VC specifier enumerated>
 <metric scale factor>
| <DEVICE VIEWPORT MAPPING>
 <isotropy flag enumerated>
 <horizontal alignment flag enumerated>
 <vertical alignment flag enumerated>
| <BACKGROUND COLOUR>
 <red green blue>
| <specification element>
| <representation element>
| <pattern table element>
| <colour table element>
| <extra element>

<specification element> => <COLOUR SELECTION MODE>
 <colour select mode enumerated>
| <LINE WIDTH SPECIFICATION MODE>
 <spec mode enumerated>
| <MARKER SIZE SPECIFICATION MODE>
 <spec mode enumerated>
| <EDGE WIDTH SPECIFICATION MODE>
 <spec mode enumerated>

<colour select mode enumerated> => <INDEXED>
| <DIRECT>

<scaling spec mode enumerated> => <ABSTRACT>
| <METRIC>

<metric scale factor> => <real>

<isotropy flag enumerated> ::= <NOT FORCED>
| <FORCED>

<horiz align flag enumer> => <LEFT>
| <CENTRE>
| <RIGHT>

<vert align flag enumer> => <BOTTOM>
| <CENTRE>
| <TOP>

<spec mode enumerated> => <ABSOLUTE>
| <SCALED>

<viewport point> ::= <vp>

<vp>	:=	<integer> (2) <real> (2)
<VC specifier enumerated>	:=	<FRACTION OF DEFAULT DRAWING SURFACE> <MILLIMETRES WITH SCALE FACTOR> <PHYSICAL DEVICE UNITS>
<representation element>	:=	<LINE REPRESENTATION> <positive index> <index> <size value> <colour> <MARKER REPRESENTATION> <positive index> <index> <size value> <colour> <TEXT REPRESENTATION> <positive index> <positive index> (font) <text precision enumerated> <real> (character spacing) <real> (expansion factor) <colour> <FILL REPRESENTATION> <positive index> <interior style enumerated> <colour> <index> (hatch index) <positive index> (pattern index) <EDGE REPRESENTATION> <positive index> <index> <size value> <colour>
<size value>	:=	<non-negative vdc value> <non-negative real>
<non-negative vdc value>	:=	<vdc value> (greater than or equal to 0)
<non-negative real>	:=	<real> (greater than or equal to 0)
<colour>	:=	<colour index> <red green blue>
<text precision enumerated>	:=	<STRING> <CHARACTER> <STROKE>
<interior style enumerated>	:=	<HOLLOW> <SOLID> <PATTERN> <HATCH> <EMPTY>

F.3.4 Control elements

<control element>	:=	<eligible control element> <BEGIN FIGURE> <END FIGURE> <NEW REGION>
-------------------	----	--

<eligible control element>

- => <vdc precision>
 - | <AUXILIARY COLOUR>
 - <colour>
 - | <TRANSPARENCY>
 - <on-off indicator enumerated>
 - | <CLIP RECT ANGLE>
 - <point>(2)
 - | <CLIP INDICATOR>
 - <on-off indicator enumerated>
 - | <LINE CLIPPING MODE>
 - <clip mode enumerated>
 - | <MARKER CLIPPING MODE>
 - <clip mode enumerated>
 - | <EDGE CLIPPING MODE>
 - <clip mode enumerated>
 - | <SAVE PRIMITIVE CONTEXT>
 - <context name>
 - | <RESTORE PRIMITIVE CONTEXT>
 - <context name>

<on-off indicator enumerated>

- => <ON>
- | <OFF>

<vdc precision>

- => <VDC INTEGER PRECISION>
 - <vdc integer precision value>
- | <VDC REAL PRECISION>
 - <vdc real precision value>
 - (these elements have encoding)
 - (dependent parameters)

<clip mode enumerated>

- => <LOCUS>
- | <SHAPE>
- | <LOCUS_THEN_SHAPE>

<context name>

- => <name>

F.3.5 Graphical elements

<graphical element>

- => <polypoint element>
- | <text element>
- | <cell element>
- | <gdp element>
- | <rectangle element>
- | <circular element>
- | <elliptical element>
- | <pointless element>

<polypoint element>

- => <POLYLINE>
 - <point pair>
 - <point list>
- | <DISJOINT POLYLINE>
 - <point pair>
 - <point pair list>
- | <POLYMARKER>
 - <point>
 - <point list>
- | <POLYGON>
 - <point>(3)
 - <point list>
- | <POLYGON SET>
 - <point edge pair>(3)
 - <point edge pair list>

<point list>	= <point>*
<point pair list>	= <point pair>*
<point pair>	= <point>(2)
<point edge pair>	= <point><edge out flag>
<point edge pair list>	= <point edge pair>*
<edge out flag>	= <INVISIBLE> <VISIBLE> <CLOSE_INVISIBLE> <CLOSE_VISIBLE>
<text element>	= <TEXT> <point> <text tail> <restricted text element>
<restricted text element>	= <RESTRICTED TEXT> <extend> <point> <text tail>
<extend>	= <dc value>(2)
<text tail>	= <final character list> <nonfinal character list>
<final character list>	= <FINAL> <string>
<nonfinal character list>	= <NOT FINAL> <string> <partial text attribute element>*
<spanned text>	= <APPEND TEXT> <text tail>
<cell element>	= <CELL ARRAY> <point>(3) <integer>(2) <local colour precision> <colour>(integer1 x integer2) (this element has an encoding) (dependent parameter)
<local colour precision>	= <colour precision value> <colour index precision value> <default colour precision indicator>
<gdp element>	= <GDP> <gdp identifier> <point list> <data record>
<gdp identifier>	= <integer>
<rectangle element>	= <RECTANGLE> <point pair>

<circular element>	=> <CIRCLE> <point> <radius> <CIRCULAR ARC 3 POINT> <point>(3) <CIRCULAR ARC 3 POINT CLOSE> <point>(3) <close type> <CIRCULAR ARC CENTRE> <point> <vdc value>(4) <radius> <CIRCULAR ARC CENTRE CLOSE> <point> <vdc value>(4) <radius> <close type> <CIRCULAR ARC CENTRE REVERSED> <point> <vdc value>(4) <radius>
<radius>	=> <non-negative vdc value>
<close type>	=> <PIE> <CHORD>
<elliptical element>	=> <ELLIPSE> <point>(3) <ELLIPTICAL ARC> <point>(3) <vdc value>(4) <ELLIPTICAL ARC CLOSE> <point>(3) <vdc value>(4) <close type>
<pointless element>	=> <CONNECTING EDGE>

F.3.6 Attribute elements

<primitive attribute element>	=> <line attribute element> <marker attribute element> <text attribute element> <filled-area attribute element> <aspect source flags> <pick identifier>
<line attribute element>	=> <LINE BUNDLE INDEX> <positive index> <LINE TYPE> <index> <LINE WIDTH> <size value> <LINE COLOUR> <colour>
<marker attribute element>	=> <MARKER BUNDLE INDEX> <positive index> <MARKER TYPE> <index> <MARKER SIZE>

	<ul style="list-style-type: none"> <size value> <MARKER COLOUR> <colour>
<text attribute element>	<ul style="list-style-type: none"> = <TEXT FONT INDEX> <positive index> <TEXT PRECISION> <text precision enumerated> <CHARACTER EXPANSION FACTOR> <real> <CHARACTER SPACING> <real> <TEXT COLOUR> <colour> <CHARACTER HEIGHT> <non-negative vdc value> <CHARACTER SET INDEX> <positive index> <ALTERNATE CHARACTER SET INDEX> <positive index> <TEXT BUNDLE INDEX> <positive index> <AUXILIARY COLOUR> <colour> <TRANSPARENCY> <on-off indicator enumerated>
<text attribute element>	<ul style="list-style-type: none"> = <char attribute element> <string attribute element>
<char attribute element>	<ul style="list-style-type: none"> = <TEXT BUNDLE INDEX> <positive index> <TEXT FONT INDEX> <positive index> <CHARACTER EXPANSION FACTOR> <real> <CHARACTER SPACING> <real> <TEXT COLOUR> <colour> <CHARACTER HEIGHT> <non-negative vdc value> <CHARACTER ORIENTATION> <vdc value>(4) <CHARACTER SET INDEX> <positive index> <ALTERNATE CHARACTER SET INDEX> <positive index>
<string attribute element>	<ul style="list-style-type: none"> = <TEXT PATH> <path enumerated> <TEXT PRECISION> <text precision enumerated> <TEXT ALIGNMENT> <horizontal align enumerated> <vertical align enumerated> <continuous align value> (2)
<path enumerated>	<ul style="list-style-type: none"> = <RIGHT> <LEFT> <UP> <DOWN>
<horizontal align enumerated>	<ul style="list-style-type: none"> = <NORMAL HORIZONTAL>

	<ul style="list-style-type: none"> <LEFT> <CENTRE> <RIGHT> <CONTINUOUS HORIZONTAL>
<vertical align enumerated>	<ul style="list-style-type: none"> = <NORMAL VERTICAL> <TOP> <CAP> <HALF> <BASE> <BOTTOM> <CONTINUOUS VERTICAL>
<continuous align value>	= <real>
<filled-area attribute elem>	<ul style="list-style-type: none"> = <FILL BUNDLE INDEX> <positive index> <INTERIOR STYLE> <interior style enumerated> <FILL COLOUR> <colour> <HATCH INDEX> <index> <PATTERN INDEX> <positive index> <EDGE BUNDLE INDEX> <positive index> <EDGE TYPE> <index> <EDGE WIDTH> <size value> <EDGE COLOUR> <colour> <EDGE VISIBILITY> <on-off indicator enumerated> <FILL REFERENCE POINT> <point> <pattern table element> <PATTERN SIZE> <vdc value>(4)
<colour table element>	<ul style="list-style-type: none"> = <COLOUR TABLE> <starting index> <red green blue>→
<pattern table element>	<ul style="list-style-type: none"> = <PATTERN TABLE> <positive index> <integer>(2) <local colour precision> <colour>(integer1 x integer2) (this element has an encoding) (dependent parameter)
<starting index>	= <colour index>
<aspect source flags>	<ul style="list-style-type: none"> = <ASPECT SOURCE FLAGS> <asf pair>→
<asf pair>	<ul style="list-style-type: none"> = <asf type enumerated> <asf enumerated>
<asf type enumerated>	<ul style="list-style-type: none"> = <LINE TYPE ASF> <LINE WIDTH ASF> <LINE COLOUR ASF>

	<MARKER TYPE ASF>
	<MARKER SIZE ASF>
	<MARKER COLOUR ASF>
	<TEXT FONT ASF>
	<TEXT PRECISION ASF>
	<CHARACTER EXPANSION FACTOR ASF>
	<CHARACTER SPACING ASF>
	<TEXT COLOUR ASF>
	<INTERIOR STYLE ASF>
	<FILL COLOUR ASF>
	<HATCH INDEX ASF>
	<PATTERN INDEX ASF>
	<EDGE TYPE ASF>
	<EDGE WIDTH ASF>
	<EDGE COLOUR ASF>
<asf enumerated>	:= <INDIVIDUAL>
	<BUNDLED>

<pick identifier>	:= <PICK IDENTIFIER>
	<name>

F.3.7 Closed figure element

<closed figure>	:= <BEGIN FIGURE>
	<eligible elements within closed figures>
	<END FIGURE>

<eligible elements within closed figures>	:=	<VDC REAL PRECISION>
		<VDC INTEGER PRECISION>
		<AUXILIARY COLOUR>
		<TRANSPARENCY>
		<END FIGURE>
		<NEW REGION>
		<POLYLINE>
		<DISJOINT POLYLINE>
		<POLYGON>
		<POLYGON SET>
		<GDP>
		<RECTANGLE>
		<CIRCLE>
		<CIRCULAR ARC 3 POINT>
		<CIRCULAR ARC 3 POINT CLOSE>
		<CIRCULAR ARC CENTRE>
		<CIRCULAR ARC CENTRE CLOSE>
		<CIRCULAR ARC CENTRE REVERSED>
		<ELLIPSE>
		<ELLIPTICAL ARC>
		<ELLIPTICAL ARC CLOSE>
		<EDGE BUNDLE INDEX>
		<EDGE TYPE>
		<EDGE WIDTH>
		<EDGE COLOUR>
		<EDGE VISIBILITY>
		<EDGE ASPECT SOURCE FLAGS>
		<ESCAPE>
		<MESSAGE>
		<APPLICATION DATA>

F.3.8 Escape elements

<escape element>	= <ESCAPE>
------------------	------------

	<identifier> <data record>
<identifier>	= <integer>
F.3.9 External elements	
<external element>	= <MESSAGE> <action flag> <string> <APPLICATION DATA> <integer> <data record>
<action flag>	= <YES> <NO>
F.3.10 Segment elements	
<segment control element>	= <COPY SEGMENT> <segment identifier> <copy transformation matrix> <segment transformation application> <INHERITANCE FILTER> <filter selection list enumerated> <setting enumerated> <CLIP INHERITANCE> <clip inheritance enumerated>
<segment attribute element>	= <SEGMENT TRANSFORMATION> <segment identifier> <transformation matrix> <SEGMENT HIGHLIGHTING> <segment identifier> <highlighting enumerated> <SEGMENT DISPLAY PRIORITY> <segment identifier> <segment display priority> <SEGMENT PICK PRIORITY> <segment name> <segment pick priority>
<segment identifier>	= <name>
<copy transformation matrix>	= <transformation matrix>
<transformation matrix>	= <2x2 matrix of reals> <2x1 matrix of vdc's>
<segment trans. application>	= <NO> <YES>
<filter selection list enumerated>	= <attribute and control name enumerated> <attribute group and control enumerated> <ASF name enumerated> <ASF group enumerated>
<attribute and control name enumerated>	= <LINE BUNDLE INDEX> <LINE TYPE>

	<ul style="list-style-type: none"> <LINE WIDTH> <LINE COLOUR> <LINE CLIPPING MODE> <MARKER BUNDLE INDEX> <MARKER TYPE> <MARKER SIZE> <MARKER COLOUR> <MARKER CLIPPING MODE> <TEXT BUNDLE INDEX> <TEXT FONT INDEX> <TEXT PRECISION> <CHARACTER EXPANSION FACTOR> <CHARACTER SPACING> <TEXT COLOUR> <CHARACTER HEIGHT> <CHARACTER ORIENTATION> <TEXT PATH> <TEXT ALIGNMENT> <FILL BUNDLE INDEX> <INTERIOR STYLE> <FILL COLOUR> <HATCH INDEX> <EDGE BUNDLE INDEX> <EDGE TYPE> <EDGE WIDTH> <EDGE COLOUR> <EDGE VISIBILITY> <EDGE CLIPPING MODE> <FILL REFERENCE POINT> <PATTERN TABLE> <PATTERN SIZE> <AUXILIARY COLOUR> <TRANSPARENCY>
<attribute group enumerated>	<ul style="list-style-type: none"> <LINE ATTRIBUTES> <MARKER ATTRIBUTES> <TEXT ATTRIBUTES> <CHARACTER ATTRIBUTES> <FILL ATTRIBUTES> <EDGE ATTRIBUTES> <PATTERN ATTRIBUTES> <OUTPUT CONTROL> <PICK IDENTIFIER> <ALL ATTRIBUTES> <ALL>
<setting enumerated>	<ul style="list-style-type: none"> <STATE_LIST> <SEGMENT>
<asf name enumerated>	<ul style="list-style-type: none"> <LINE TYPE ASF> <LINE WIDTH ASF> <LINE COLOUR ASF> <MARKER TYPE ASF> <MARKER SIZE ASF> <MARKER COLOUR ASF> <TEXT FONT INDEX ASF> <TEXT PRECISION ASF> <CHARACTER EXPANSION FACTOR ASF> <CHARACTER SPACING ASF> <TEXT COLOUR ASF> <INTERIOR STYLE ASF> <FILL COLOUR ASF> <HATCH INDEX ASF>

	<PATTERN INDEX ASF>
	<EDGE TYPE ASF>
	<EDGE WIDTH ASF>
	<EDGE COLOUR ASF>
<asf group enumerated>	:= <LINE ASF>
	<MARKER ASFS>
	<TEXT ASFS>
	<FILL ASFS>
	<EDGE ASFS>
	<ALL ASFS>
<clip inheritance enumerated>	:= <STATE_LIST>
	<INTERSECTION>
<highlighting enumerated>	:= <NORMAL>
	<HIGHLIGHTED>
<segment display priority>	::= <integer>
<segment pick priority>	::= <integer>

F.4 Terminal symbols

The following are the terminals in this grammar.
 Their representation is dependent on the encoding scheme used.
 In annex A of the subsequent parts of this Standard, these
 encoding-dependent symbols are further described.

<element name>
 <integer>
 <real>
 <vdc value>
 <string>
 <colour index>
 <red green blue>
 <integer precision value>
 <real precision value>
 <index precision value>
 <colour precision value>
 <col index precision value>
 <name precision value>
 <default col precision indicator>
 <vdc integer precision value>
 <vdc real precision value>
 <data record>
 <name>
 <viewport point>
 <2x2 matrix of reals>
 <2x1 matrix of vdc>

The CGM extended opcodes are encoding dependent. A complete list of them can be found in the productions for <element name enumerated> below.

The enumerated types:

<INTEGER>
 <REAL>
 <ON>
 <OFF>
 <INDEXED>

<DIRECT>
 <ABSTRACT>
 <METRIC>
 <ABSOLUTE>
 <SCALED>
 <94 CHAR>
 <96 CHAR>
 <MULTI-BYTE 94 CHAR>
 <MULTI-BYTE 96 CHAR>
 <COMPLETE CODE>
 <BASIC 7-BIT>
 <BASIC 8-BIT>
 <EXTENDED 7-BIT>
 <EXTENDED 8-BIT>
 <FRACTION OF DEFAULT DEVICE VIEWPORT>
 <MILLIMETRES WITH SCALE FACTOR>
 <PHYSICAL DEVICE UNITS>
 <NOT FORCED>
 <FORCED>
 <LEFT>
 <RIGHT>
 <CENTRE>
 <BOTTOM>
 <TOP>
 <LOCUS>
 <SHAPE>
 <LOCUS_THEN_SHAPE>
 <INVISIBLE>
 <VISIBLE>
 <CLOSE_INVISIBLE>
 <CLOSE_VISIBLE>
 <PIE>
 <CHORD>
 <FINAL>
 <NOT FINAL>
 <INDIVIDUAL>
 <BUNDLED>
 <HOLLOW>
 <SOLID>
 <PATTERN>
 <HATCH>
 <EMPTY>
 <STRING>
 <CHARACTER>
 <STROKE>
 <UP>
 <DOWN>
 <NORMAL HORIZONTAL>
 <CONTINUOUS HORIZONTAL>
 <NORMAL VERTICAL>
 <CAP>
 <HALF>
 <BASE>
 <CONTINUOUS VERTICAL>
 <YES>
 <NO>
 <LINE TYPE ASF>
 <LINE WIDTH ASF>
 <LINE COLOUR ASF>
 <MARKER TYPE ASF>
 <MARKER SIZE ASF>
 <MARKER COLOUR ASF>
 <TEXT FONT ASF>
 <TEXT PRECISION ASF>

<CHARACTER EXPANSION FACTOR ASF>
 <CHARACTER SPACING ASF>
 <TEXT COLOUR ASF>
 <INTERIOR STYLE ASF>
 <HATCH INDEX ASF>
 <PATTERN INDEX ASF>
 <FILL COLOUR ASF>
 <EDGE TYPE ASF>
 <EDGE WIDTH ASF>
 <EDGE COLOUR ASF>
 <LINE ATTRIBUTES>
 <MARKER ATTRIBUTES>
 <TEXT ATTRIBUTES>
 <CHARACTER ATTRIBUTES>
 <FILL ATTRIBUTES>
 <EDGE ATTRIBUTES>
 <PATTERN ATTRIBUTES>
 <OUTPUT CONTROL>
 <ALL ATTRIBUTES>
 <ALL>
 <LINE BUNDLE INDEX>
 <LINE TYPE>
 <LINE WIDTH>
 <LINE COLOUR>
 <LINE CLIPPING MODE>
 <MARKER BUNDLE INDEX>
 <MARKER TYPE>
 <MARKER SIZE>
 <MARKER COLOUR>
 <MARKER CLIPPING MODE>
 <TEXT BUNDLE INDEX>
 <TEXT FONT INDEX>
 <TEXT PRECISION>
 <CHARACTER EXPANSION FACTOR>
 <CHARACTER SPACING>
 <TEXT COLOUR>
 <CHARACTER HEIGHT>
 <CHARACTER ORIENTATION>
 <TEXT PATH>
 <TEXT ALIGNMENT>
 <CHARACTER SET INDEX>
 <ALTERNATE CHARACTER SET INDEX>
 <FILL BUNDLE INDEX>
 <INTERIOR STYLE>
 <FILL COLOUR>
 <HATCH INDEX>
 <PATTERN INDEX>
 <EDGE BUNDLE INDEX>
 <EDGE TYPE>
 <EDGE WIDTH>
 <EDGE COLOUR>
 <EDGE VISIBILITY>
 <EDGE CLIPPING MODE>
 <FILL REFERENCE POINT>
 <PATTERN SIZE>
 <AUXILIARY COLOUR>
 <TRANSPARENCY>
 <STATE_LIST>
 <INTERSECTION>
 <SEGMENT>
 <LINE ASFS>
 <MARKER ASFS>
 <TEXT ASFS>
 <FILL ASFS>

<EDGE ASFS>
 <ALL ASFS>
 <NORMAL>
 <HIGHLIGHTED>

<element name enumerated>

= <BEGIN METAFILE>
 | <END METAFILE>
 | <BEGIN PICTURE>
 | <BEGIN PICTURE BODY>
 | <END PICTURE>
 | <BEGIN SEGMENT>
 | <END SEGMENT>
 | <METAFILE VERSION>
 | <METAFILE DESCRIPTION>
 | <VDC TYPE>
 | <INTEGER PRECISION>
 | <REAL PRECISION>
 | <INDEX PRECISION>
 | <COLOUR PRECISION>
 | <COLOUR INDEX PRECISION>
 | <NAME PRECISION>
 | <MAXIMUM COLOUR INDEX>
 | <COLOUR VALUE EXTENT>
 | <METAFILE ELEMENT LIST>
 | <METAFILE DEFAULTS REPLACEMENT>
 |
 | <CHARACTER SET LIST>
 | <CHARACTER CODING ANNOUNCER>
 | <MAXIMUM VDC EXTENT>
 | <SEGMENT PRIORITY EXTENT>
 | <SCALING MODE>
 | <COLOUR SELECTION MODE>
 | <LINE WIDTH SPECIFICATION MODE>
 | <MARKER SIZE SPECIFICATION MODE>
 | <EDGE WIDTH SPECIFICATION MODE>
 | <VDC EXTENT>
 | <BACKGROUND COLOUR>
 | <DEVICE VIEWPORT>
 | <DEVICE VIEWPORT SPECIFICATION MODE>
 | <DEVICE VIEWPORT MAPPING>
 | <LINE REPRESENTATION>
 | <MARKER REPRESENTATION>
 | <TEXT REPRESENTATION>
 | <FILL REPRESENTATION>
 | <EDGE REPRESENTATION>
 | <VDC INTEGER PRECISION>
 | <VDC REAL PRECISION>
 | <AUXILIARY COLOUR>
 | <TRANSPARENCY>
 | <CLIP RECTANGLE>
 | <CLIP INDICATOR>
 | <LINE CLIPPING MODE>
 | <MARKER CLIPPING MODE>
 | <EDGE CLIPPING MODE>
 | <BEGIN FIGURE>
 | <END FIGURE>
 | <NEW REGION>
 | <SAVE PRIMITIVE CONTEXT>
 | <RESTORE PRIMITIVE CONTEXT>
 | <POLYLINE>
 | <DISJOINT POLYLINE>
 | <POLYMARKER>
 | <TEXT>

<RESTRICTED TEXT>
 <APPEND TEXT>
 <POLYGON>
 <POLYGON SET>
 <CELL ARRAY>
 <GDP>
 <RECTANGLE>
 <CIRCLE>
 <CIRCULAR ARC 3 POINT>
 <CIRCULAR ARC 3 POINT CLOSE>
 <CIRCULAR ARC CENTRE>
 <CIRCULAR ARC CENTRE CLOSE>
 <CIRCULAR ARC CENTRE REVERSED>
 <ELLIPSE>
 <ELLIPTICAL ARC>
 <ELLIPTICAL ARC CLOSE>
 <CONNECTING EDGE>
 <LINE BUNDLE INDEX>
 <LINE TYPE>
 <LINE WIDTH>
 <LINE COLOUR>
 <MARKER BUNDLE INDEX>
 <MARKER TYPE>
 <MARKER SIZE>
 <MARKER COLOUR>
 <TEXT BUNDLE INDEX>
 <TEXT FONT INDEX>
 <TEXT PRECISION>
 <CHARACTER EXPANSION FACTOR>
 <CHARACTER SPACING>
 <TEXT COLOUR>
 <CHARACTER HEIGHT>
 <CHARACTER ORIENTATION>
 <TEXT PATH>
 <TEXT ALIGNMENT>
 <CHARACTER SET INDEX>
 <ALTERNATE CHARACTER SET INDEX>
 <FILL BUNDLE INDEX>
 <INTERIOR STYLE>
 <FILL COLOUR>
 <HATCH INDEX>
 <PATTERN INDEX>
 <EDGE BUNDLE INDEX>
 <EDGE TYPE>
 <EDGE WIDTH>
 <EDGE COLOUR>
 <EDGE VISIBILITY>
 <FILL REFERENCE POINT>
 <PATTERN TABLE>
 <PATTERN SIZE>
 <COLOUR TABLE>
 <ASPECT SOURCE FLAGS>
 <PICK IDENTIFIER>
 <COPY SEGMENT>
 <INHERITANCE FILTER>
 <CLIP INHERITANCE>
 <CLIP INHERITANCE>
 <SEGMENT TRANSFORMATION>
 <SEGMENT HIGHLIGHTING>
 <SEGMENT DISPLAY PRIORITY>
 <SEGMENT PICK PRIORITY>
 <PIXEL ARRAY>
 <ESCAPE>
 <MESSAGE>

| <APPLICATION DATA>
| <DRAWING SET>
| <DRAWING PLUS CONTROL SET>
| <VER.2 STATIC ALL SET>
| <EXTENDED PRIMITIVES SET>
| <VER.2 STATIC GKSM SET>

Annex G Relationship of CGM and GKS

(This annex does not form a part of the standard)

G.1 Introduction

The GKS Standard includes the concepts of metafile input and output workstations as well as functions providing access to and interpretation of metafiles. It does not, however, contain a metafile definition as part of the Standard. Annex E of this standard provides a mapping to version 1 metafiles.

This Annex provides a mapping between GKS and the version 2 metafiles.

G.2 Scope

The CGM Add.2 captures static picture definitions. GKS provides many possibilities to generate images. This means that the strategies for generating picture definitions are numerous and complex. The best strategy to use in given circumstances is dictated by implementation and application requirements. This annex presents a detailed mappings between GKS and CGM only for one particular strategy.

The scope of this annex is further limited to generation of metafiles by GKS and interpretation of GKS generated metafiles in GKS environments. There are many other scenarios for generation and interpretation of metafiles, such as interpretation by GKS of metafiles not generated by GKS and interpretation by non-GKS processes of GKS generated metafiles. These scenarios are not dealt with in this annex. The annex C presents context models dealing with such cases.

G.3 Overview of the Differences Between GKS and CGM Version 2

While CGM supports all of the basic output functionality of GKS, a one-to-one mapping between GKS and CGM is not possible in all cases mainly because some CGM elements have no counterparts as GKS functions and some GKS functions have no corresponding CGM element. Examples of this are:

1. Delimiter element like BEGIN PICTURE
2. Enhanced facilities for tailoring and controlling the interpretation of the metafile precision of various items, and the control of default values.
3. Extended capabilities in the area of text processing, such as named font, changing character sets and restricted text.

G.4 Mapping Concepts

The tables later in this annex present mappings between GKS and CGM elements.

G.4.1 Principles

The following principles are basis of the GKS/CGM model of this annex and of the function mappings themselves:

- a) conceptual compatibility with GKS
- b) compatibility with the design concepts of CGM

G.4.2 Workstation

The CGM is generated, in this model, by a workstation of type MO. The behaviour of the workstation, particularly in response to dynamic GKS functions, can be illustrated by analogy: in most respects, the MO/CGM workstation in GKS may be implemented in a manner analogous to a workstation of category OUTPUT (e.g., a plotter), whose device instruction set corresponds to the CGM elements. Strategies for correctly sending device instructions to such a real device are similar to those generating the proper elements on the metafile.

The CGM is read by a workstation of category MI. Certain elements, such as the metafile descriptor and precision-setting elements, are viewed as directives to the MI workstation itself, so that it may correctly read the metafile contents.

G.4.3 Picture generation

A metafile is composed of a collection of mutually independent pictures. GKS does not have the concept of "picture" as defined in CGM but it does formalize the notion of an empty view surface. GKS actions which cause clearing of the view surface, such as CLEAR WORKSTATION, are defined to delimit metafile picture. There is another mechanism which leads to generation of pictures in this model of the GKS/CGM relationship. GKS contains functions which have potential dynamic effects on a non-empty display surface. The CGM design concepts exclude dynamic modification of pictures. For this reason all "dynamic modification accepted ..." values of a MO/CGM workstation will be conceptually DRG.

The default value of the deferral state on an MO/CGM workstation is ASTI-SUPPRESSED

This model of the MO/CGM workstation defines that whenever a GKS function is invoked which causes a regeneration, then a picture is output to the metafile.

G.4.4 Coordinates and clipping

The coordinate space of the metafile, VDC, is defined as being identical to the NDC space of GKS. Clipping and transformation are completely deferred to the metafile interpreter. Each GKS clip and transformation element has a counterpart in CGM.

G.4.5 Workstation transformation

The workstation transformation is defined in GKS by setting a workstation window in device independent NDC and a workstation viewport in device-dependent DC. The workstation window is written to the metafile with the VDC EXTENT element. The workstation viewport is written to the metafile with the DEVICE VIEWPORT element.

The default values of DEVICE VIEWPORT MAPPING correspond to GKS mapping of the device coordinate system onto the display space. The DEVICE VIEWPORT SPECIFICATION MODE is set to MILLIMETRES WITH SCALE FACTOR and metric scale factor 1000.0 within the METAFILE DEFAULTS REPLACEMENT element.

G.4.7 Metafile element list

The metafile element list short hand defined for use with GKS application is 'version2-static gksm'

G.4.8 Relationship of fonts between CGM and GKS

The GKS standard includes the concepts of text output primitive attributes. However, the mechanism for specifying the text font differs from that specified in the CGM standard. This clause defines the approach to handling these attributes within the GKS environment.

G.4.8.1 Overview of the differences between GKS and CGM fonts

While CGM supports the TEXT output primitive attribute functionality of GKS, a one-to-one mapping between CGM and GKS is not possible in all cases. Specifically:

- 1) GKS and CGM differ in the way fonts are defined:

In the CGM text fonts are defined with the FONT LIST element that associates font names or identifications with entries in a Font Table.

In GKS, no mechanism is available for defining text fonts. GKS associates a unique text font number with each font. The Registration Authority is responsible for defining this mapping of font numbers to specific font identifications.

- 2) GKS and CGM differ in the way fonts are selected:

In the CGM, text fonts are selected with the TEXT FONT INDEX element. The index selects an individual font from different fonts in the font list.

In GKS, text fonts are selected with a font number. The font number selects a specific GKS registered font if the value is positive. If the font number is negative an implementation dependent font is selected.

- 3) GKS and CGM differ on the independence of font and text precision:

In the CGM, the font and text precision are specified by independent elements.

In GKS, the font and text precision are directly associated with specification by a single function.

4) Some CGM Elements have no counterpart as GKS functions:

These include Auxiliary Colour related elements, such as AUXILIARY COLOUR and TRANSPARENCY that affect the presentation of text.

This additional functionality of the CGM causes no special problems for a GKS environment interpreting a version 2 CGM.

5) The character set related elements CHARACTER SET LIST, CHARACTER CODING ANNOUNCER, CHARACTER SET INDEX, ALTERNATE CHARACTER SET INDEX have no counterpart in GKS. GKS does not recognize the concept of character set as a separate concept from the font concept. GKS implementors are encouraged to provide a mapping to the character set elements for both MO and MI workstations to increase the possibility of transferring metafile between GKS environments and other systems.

G.4.8.2 Suggestion for interpretation of CGM font information by GKS

GKS environments interpreting a CGM specify fonts with a font number. It is assumed that GKS maintains a list associating positive font numbers with a GKS registered font name or identifier. Private font numbers (i.e. negative values) must be maintained in an implementation dependent list of associations. As the FONT LIST element is interpreted, an additional list must be maintained that associates individual font names specified in the CGM with a font index. When the TEXT FONT INDEX element is interpreted, the font name associated with the font index is determined from the list of currently used fonts. The font name is used to determine the GKS font number associated with this font from a list of GKS registered fonts. This font number is used as the font parameter of the TEXT FONT AND PRECISION function. The value of the precision parameter is taken from the TEXT PRECISION element.

G.4.8.3 Generating CGM font information from GKS

When generating font information from GKS via TEXT FONT AND PRECISION it is recommended that the generator also writes the elements CHARACTER SET INDEX and ALTERNATE CHARACTER SET INDEX as well as TEXT FONT INDEX and TEXT PRECISION. The generator is assumed to have a table associating the positive font numbers of GKS with the registered names. The generator shall put a FONT LIST element in the Metafile Descriptor with the names of those fonts referenced by positive GKS font numbers. Negative GKS font numbers are private and must be mapped to CGM font indices which are positive and beyond the range of the table.

NOTE: The metafile must be completely generated before the FONT LIST element can be written.

G.5 Metafile Generation

Included in following tables is a particular set of mappings of the GKS function, workstation state list entries and segment state list entries onto CGM elements. The mappings presented are deemed usable and suitable for guiding implementation of a CGM picture generator in a GKS environment. The mapping concepts of G.4 are assumed.

G.5.1 Control functions

GKS function	CGM Add.2 elements	Notes
OPEN WORKSTATION	BEGIN METAFILE	(1)
	(Metafile Descriptor)	(2)
	BEGIN PICTURE	(3)
	store current workstation state list	(4)
	BEGIN PICTURE BODY	
CLOSE WORKSTATION	END PICTURE	
	END METAFILE	
ACTIVATE WORKSTATION	attribute settings CLIP RECTANGLE	(5)

	CLIP INDICATOR	
	enable output to metafile	
DEACTIVATE WORKSTATION	disable output to metafile	
CLEAR WORKSTATION	no Action	
control flag = CONDITIONAL		
display space empty = EMPTY		
CLEAR WORKSTATION	END PICTURE	
display space empty = NOTEMPTY	BEGIN PICTURE	(3)
	store current workstation state list	(4)
	BEGIN PICTURE BODY	
	attribute settings	(5)
	CLIP RECTANGLE	
	CLIP INDICATOR	
REDRAW ALL SEGMENTS ON WORKSTATION	no Action	
display space empty = EMPTY		
REDRAW ALL SEGMENTS ON WORKSTATION	END PICTURE	
display space empty = NOTEMPTY	BEGIN PICTURE	(3)
	store current workstation state list	(4)
	BEGIN PICTURE BODY	
	attribute settings	(5)
	CLIP RECTANGLE	
	CLIP INDICATOR	
	generate all visible segments stored for	
	the MO workstation	(6)
UPDATE WORKSTATION	as REDRAW ALL SEGMENTS ON WORKSTATION	
regeneration flag = PERFORM		
new frame action necessary		
at update = YES		
UPDATE WORKSTATION		
regeneration flag = PERFORM		
new frame action necessary		
at update = NO		
or		
UPDATE WORKSTATION	no Action	
regeneration flag = POSTPONE		
SET DEFERRAL STATE	as REDRAW ALL SEGMENTS ON WORKSTATION	
new frame action necessary		
at update = YES		
ESCAPE	ESCAPE	
MESSAGE	MESSAGE	

NOTES

- 1) The use of the identifier parameter in BEGIN METAFILE is implementation dependent.
- 2) See G.5.5 Metafile Descriptor
- 3) The use of the identifier parameter in BEGIN PICTURE is implementation dependent.
- 4) See G.5.6 mapping of workstation state list to CGM element.
- 5) The attribute settings ensure that the metafile attributes in effect when the first graphical primitive element of a picture is encountered match the current attributes from the GKS state list.
- 6) Generated sequence of CGM-Elements for every segment as ASSOCIATE SEGMENT WITH WORKSTATION (see G.5.3.4)

G.5.2 GKS function leading to an implicit regeneration

Depending on the deferral state the following GKS functions may act as REDRAW ALL SEGMENTS ON WORKSTATION because conceptually all corresponding "dynamic modification accepted ..." entries in the workstation description table are set to IRG (see G.4.3)

SET POLYLINE REPRESENTATION
SET MARKER REPRESENTATION
SET TEXT REPRESENTATION
SET INTERIOR REPRESENTATION
SET PATTERN REPRESENTATION
SET COLOUR REPRESENTATION

SET WORKSTATION WINDOW
SET WORKSTATION VIEWPORT

SET SEGMENT TRANSFORMATION
SET VISIBILITY
SET HIGHLIGHTING
SET SEGMENT PRIORITY

all primitives added to open segments overlapping segments of higher priority.

DELETE SEGMENT
DELETE SEGMENT FROM WORKSTATION
ASSOCIATE SEGMENT WITH WORKSTATION

G.5.3 GKS function with no direct dynamic effect

G.5.3.1 Mappings of output function

GKS function	CGM Element	Notes
POLYLINE	POLYLINE	
POLYMARKER	POLYMARKER	
TEXT	TEXT	
FILL AREA	FILL AREA	
CELL ARRAY	CELL ARRAY	
GDP	GDP	
.		

G.5.3.2 Mappings of output attributes

GKS function	CGM element	Notes
SET POLYLINE INDEX	LINE BUNDLE INDEX	
SET LINETYPE	LINE TYPE	
SET LINewidth SCALE FACTOR	LINE WIDTH	(1)
SET POLYLINE COLOUR INDEX	LINE COLOUR	(2)
SET POLYMARKER INDEX	MARKER BUNDLE INDEX	
SET MARKER TYPE	MARKER TYPE	
SET MARKER SIZE SCALE FACTOR	MARKER SIZE	(1)
SET POLYMARKER COLOUR INDEX	MARKER COLOUR	(2)
SET TEXT INDEX	TEXT BUNDLE INDEX	
SET TEXT FONT AND PRECISION	TEXT FONT INDEX	(3)
	TEXT PRECISION	
	CHARACTER SET INDEX	
	ALTERNATE CHARACTER SET INDEX	

SET CHARACTER EXPANSION FACTOR	CHARACTER EXPANSION FACTOR	
SET CHARACTER SPACING	CHARACTER SPACING	
SET TEXT COLOUR INDEX	TEXT COLOUR	(2)
SET CHARACTER HEIGHT	CHARACTER HEIGHT	
SET CHARACTER UP VECTOR	CHARACTER ORIENTATION	
SET TEXT PATH	TEXT PATH	
SET TEXT ALIGNMENT	TEXT ALIGNMENT	
SET FILL AREA INDEX	FILL BUNDLE INDEX	
SET FILL AREA STYLE	INTERIOR STYLE	
SET FILL AREA STYLE INDEX	HATCH INDEX	(4)
	PATTERN INDEX	
SET FILL AREA COLOUR INDEX	FILL COLOUR	
SET PATTERN SIZE	PATTERN SIZE	
SET PATTERN REFERENCE POINT	FILL REFERENCE POINT	
SET ASPECT SOURCE FLAG	ASPECT SOURCE FLAGS	
SET PICK IDENTIFIER	PICK IDENTIFIER	

NOTES

- 1) The default specification modes SCALED apply.
- 2) The default colour selection mode INDEXED applies.
- 3) GKS includes the notion of character set within 'font', whereas CGM separates the two concepts. When the value of 'font' in the GKS state list changes, then the CGM elements TEXT FONT INDEX, TEXT PRECISION, CHARACTER SET INDEX and ALTERNATE CHARACTER SET INDEX are written to the metafile, each with the value of the 'font' and 'precision' entry in the GKS state list. The CGM font index is determined as described sub-clause G.4.8.2. The elements shall appear consecutively in the metafile but may appear in any order.
- 4) Legal values of the GKS 'fill area style index' differ depending upon whether the current interior style is 'hatch' or 'pattern'. Therefore a negative GKS style index results only on the generation of the HATCH INDEX element, and a positive value results in the generation of both the HATCH INDEX and PATTERN INDEX elements.

G.5.3.3 Mappings of transformation function

GKS function	CGM Element	Notes
SET WINDOW (of current selected normalisation transformation)	CHARACTER HEIGHT CHARACTER ORIENTATION PATTERN SIZE FILL REFERENCE POINT	
SET VIEWPORT (of current selected normalisation transformation)	CHARACTER HEIGHT CHARACTER ORIENTATION PATTERN SIZE FILL REFERENCE POINT CLIP RECTANGLE	
SET NORMALISATION TRANSFORMATION	CHARACTER HEIGHT CHARACTER ORIENTATION PATTERN SIZE FILL REFERENCE POINT CLIP RECTANGLE	

G.5.3.4 Mappings of segment manipulation function

GKS function	CGM element	Notes
--------------	-------------	-------

CREATE SEGMENT
CLOSE SEGMENT
RENAME SEGMENT
ASSOCIATE SEGMENT WITH WORKSTATION

BEGIN SEGMENT
END SEGMENT
no action
BEGIN SEGMENT
(segment attributes from the
segment state list)
(primitives and their associated
attributes, clip rectangle and clip
indicator)
END SEGMENT

COPY SEGMENT TO WORKSTATION

(transformed (1)
primitives and their associated
attributes, clip rectangle and clip
indicator)

INSERT SEGMENT TO WORKSTATION

(transformed primitives and
associated attributes) (2)

NOTES

- 1) Primitives transformed by the segment transformation.
- 2) Primitives transformed by the segment transformation followed by the insert transformation.

G.5.3.5 Mappings of segment attributes

see G.5.2.4, G.5.4 and G.5.7

G.5.4 GKS function with no Action

sentence not list

GKS function

Notes

SET DETECTABILITY
SET VIEWPORT INPUT PRIORITY
all input function
all inquiry function
all utility function
all error handling function

G.5.5 Metafile Description

At the head of a metafile is a set of Metafile Descriptor (MD) elements. It is useful to view these elements as forming a Metafile Description Table (similar to the GKS and Workstation Description Table in GKS).

In the GKS context, the following description table would be written at the beginning of a metafile. For the elements which are listed as "i.d.", it is implementation dependent both whether the elements are included in the table and what values are assigned to the elements.

Metafile element list	Element Value	Mandatory
METAFILE VERSION	2	X
METAFILE DESCRIPTION	i.d.	
METAFILE ELEMENT LIST	i.d.	X
VDC TYPE	i.d.	X
INTEGER PRECISION	i.d.	
REAL PRECISION	i.d.	
INDEX PRECISION	i.d.	

COLOUR PRECISION	i.d.
COLOUR INDEX PRECISION	i.d.
MAXIMUM VDC EXTENT	i.d.
MAXIMUM NPC EXTENT	i.d.
COLOUR VALUE EXTENT	i.d.
SEGMENT PRIORITY EXTENT	i.d.
METAFILE DEFAULT REPLACEMENT	i.d.
FONT LIST	i.d.
CHARACTER CODING ANNOUNCER	i.d.
CHARACTER SET LIST	i.d.
global segments	i.d.

i.d. = implementation dependent

METAFILE VERSION, METAFILE CATEGORY and METAFILE ELEMENT LIST are mandatory. All metafile defaults satisfy the GKS Description Table. Inclusion of the METAFILE DEFAULTS REPLACEMENT element to change any control, picture descriptor, and attribute defaults is optional and implementation dependent. It is also implementation dependent whether the CGM generator includes any of the other MD elements, such as the precision setting elements.

G.5.6 Mapping of workstation state list entries to CGM elements

GKS workstation state list entry	CGM element
requested workstation window	NDC EXTENT (1)
requested workstation viewport	DEVICE VIEWPORT (2)
every entry of polyline bundle table	LINE REPRESENTATION
" " " polymarker " "	MARKER "
" " " text " "	TEXT "
" " " interior " "	FILL "
" " " pattern table	PATTERN TABLE
" " " COLOUR " "	COLOUR "

NOTES

- 1) The position of the workstation window within the NDC unit square corresponds to position of the VDC extent within the maximum VDC extent.
- 2) DEVICE VIEWPORT SPECIFICATION MODE and DEVICE VIEWPORT MAPPING may be specified only within METAFILE DEFAULTS REPLACEMENT in the metafile descriptor. The VSU specifier may be either millimetres with scale factor with metric scale factor 1000.0, or 'physical device units'.

G.5.7 Mapping of segment state list entries to CGM Add.2 elements

GKS segment state list entry	CGM element
segment transformation matrix	SEGMENT TRANSFORMATION
visibility	— (1)
highlighting	SEGMENT HIGHLIGHTING
segment priority	SEGMENT DISPLAY PRIORITY
	SEGMENT PICK PRIORITY (2)
detectability	—

NOTES

- 1) invisible segments are not mapped.
- 2) The elements shall appear consecutively in the metafile but may appear in any order.

G.5.8 Mapping of metafile function

GKS function	CGM element
WRITE ITEM TO METAFILE	APPLICATION DATA (1)

NOTES

- 1) The GKS item type is mapped to the CGM identifier.

G.6 Metafile Interpretation

G.6.1 Introduction

This sub-clause describes how metafile elements from a version 2 metafile generated by a GKS program according to the mapping described in sub-clause G.5, are subsequently interpreted by the GKS INTERPRET ITEM function and/or the MI/CGM workstation. Other guidelines for interpretation are possible.

Those CGM elements which do not map to a GKS item are viewed as directives to the MI/CGM workstation itself, so that it may correctly read the metafile contents.

A number of the elements below are specified as causing GKS state list entries to be set, and have parameters specified in VDC (which corresponds to GKS NDC). The GKS state list entries are in WC. The VDC (NDC) are mapped by the inverse of the current normalization transformation before the GKS state list values are set. The table also includes item types to be returned to GKS. These are adopted from GKS Annex E.

G.6.2 Mapping of delimiter elements

CGM Element	GKS Metafile Interface	Item	Notes
BEGIN METAFILE	-	-	(1)
END METAFILE	END ITEM	0	(2)
BEGIN PICTURE	-	-	(3)
BEGIN PICTURE BODY	CLEAR WORKSTATION	1	(4)
END PICTURE	-	-	
BEGIN SEGMENT	CREATE SEGMENT	81	
END SEGMENT	CLOSE SEGMENT	82	

NOTES

- 1) The first CGM element interpreted by the MI workstation. The metafile description table immediately follows. Its elements inform the MI workstation how to read the metafile.
- 2) No further items may be read.
- 3) Appropriate GKS state list values are set to correspond to CGM defaults. Appropriate workstation state list values on active OUTPUT and OUTIN workstations are set to correspond to CGM defaults. It is not intended that this action, or the interpretation of any picture descriptor elements, cause any immediate dynamic changes to the view surface, which is cleared upon BEGIN PICTURE BODY - the implementation may wish to buffer these actions to suppress such changes, if such changes are undesirable. Only picture descriptor elements may be interpreted until BEGIN PICTURE BODY.
- 4) Causes a CLEAR WORKSTATION on all active workstations.

G.6.3 Mapping of metafile descriptor elements

All elements in this class contain only directives to the MI workstation, their interpretation does not correspond to the invocation of any GKS function.

CGM Element	GKS Metafile Interface	Item	Notes
METAFILE VERSION	-	-	(1)
METAFILE DESCRIPTION	-	-	
VDC TYPE	-	-	

INTEGER PRECISION	-	-	
REAL PRECISION	-	-	
INDEX PRECISION	-	-	
COLOUR PRECISION	-	-	
COLOUR INDEX PRECISION	-	-	
MAXIMUM COLOUR INDEX	-	-	
COLOUR VALUE EXTENT	-	-	(2)
METAFILE ELEMENT LIST	-	-	
METAFILE DEFAULTS REPLACEMENT	-	-	
FONT LIST	-	-	
CHARACTER SET LIST	-	-	
CHARACTER CODING ANNOUNCER	-	-	
MAXIMUM VDC EXTENT	-	-	(3)
MAXIMUM NPC EXTENT	-	-	(4)
SEGMENT PRIORITY EXTENT	-	-	(5)

NOTES:

- 1) The value of the parameter must be 2.
- 2) Used to normalize colour direct values to the continuous range of real numbers [0.1].
- 3) Used to normalize VDC range (i.e. NDC) and applies to VDC type INTEGER or REAL.
- 4) Used to normalize NPC range to allow the proper position of the NPC EXTENT (workstation window) within the normalized NPC range (the unit square).
- 5) Used to normalize segment priority to the continuous range of real numbers [0.1].

G.6.4 Mapping of picture descriptor elements

CGM Element	GKS Metafile Interface	Item	Notes
COLOUR SELECTION MODE	-	-	
BACKGROUND COLOUR	-	-	
DEVICE VIEWPORT	WORKSTATION VIEWPORT	72	
DEVICE VIEWPORT SPECIFICATION MODE	-	-	(1)
DEVICE VIEWPORT MAPPING	-	-	(2)
LINE REPRESENTATION	POLYLINE REPRESENTATION	51	
MARKER REPRESENTATION	POLYMARKER REPRESENTATION	52	
TEXT REPRESENTATION	TEXT REPRESENTATION	53	
FILL REPRESENTATION	FILL AREA REPRESENTATION	54	
PATTERN TABLE	PATTERN REPRESENTATION	56	
COLOUR TABLE	COLOUR REPRESENTATION	57	

NOTES:

- 1) The VSU specifier may be either MILLIMETRES WITH SCALE FACTOR with metric scale factor equal to 1000.0, or PHYSICAL DEVICE UNITS. DEVICE VIEWPORT SPECIFICATION MODE may occur only within METAFILE DEFAULTS REPLACEMENT.
- 2) The isotropy flag must be FORCED and the alignment flags must be LEFT and BOTTOM. DEVICE VIEWPORT MAPPING may occur only within METAFILE DEFAULTS REPLACEMENT.

G.6.5. Mapping of control elements

CGM Element	GKS Metafile Interface	Item	Notes
-------------	------------------------	------	-------

VDC INTEGER PRECISION	-	-
VDC REAL PRECISION	-	-
AUXILIARY COLOUR	-	-
TRANSPARENCY	-	-
CLIP RECTANGLE	CLIPPING RECTANGLE	61
CLIP INDICATOR	CLIPPING INDICATOR	62 ***??***
SAVE PRIMITIVE ATTRIBUTES 3	-	-
RESTORE PRIMITIVE ATTRIBUTES 3	-	-
CLIP VOLUME	-	-

G.6.6 Mapping of graphical primitive elements

CGM Element	GKS Metafile Interface	Item	Notes
POLYLINE	POLYLINE	11	
DISJOINT POLYLINE	-	-	
POLYMARKER	POLYMARKER	12	
APPEND TEXT	-	-	
TEXT	TEXT	13	(1)
POLYGON	FILL AREA	14	
CELL ARRAY	CELL ARRAY	15	
GDP	GDP	16	

NOTES

- 1) The text flag should be 'final'.

G.6.7 Mapping of attribute elements

CGM Element	GKS Metafile Interface	Item	Notes
LINE BUNDLE INDEX	POLYLINE INDEX	21	
LINE TYPE	LINE TYPE	22	
LINE WIDTH	LINE WIDTH SCALE FACTOR	23	(1)
LINE COLOUR	POLYLINE COLOUR INDEX	24	(2)
MARKER BUNDLE INDEX	POLYMARKER INDEX	25	
MARKER TYPE	MARKER TYPE	26	
MARKER SIZE	MARKER SIZE SCALE FACT	27	(1)
MARKER COLOUR	POLYMARKER COL INDEX	28	(2)
TEXT BUNDLE INDEX	TEXT INDEX	29	
TEXT FONT INDEX	TEXT FONT AND PRECISION	30	(3)
TEXT PRECISION	TEXT FONT AND PRECISION	30	(3)
CHARACTER EXPANSION FACTOR	CHARACTER EXPANSION FACTOR	31	
CHARACTER SPACING	CHARACTER SPACING	32	
TEXT COLOUR	TEXT COLOUR INDEX	33	(2)
CHARACTER HEIGHT	CHARACTER HEIGHT	34	(4)
CHARACTER ORIENTATION	CHARACTER UP VECTOR	34	(4)
TEXT PATH	TEXT PATH	35	
TEXT ALIGNMENT	TEXT ALIGNMENT	36	
CHARACTER SET INDEX	TEXT FONT AND PRECISION	30	(3)
ALTERNATE CHARACTER SET INDEX	TEXT FONT AND PRECISION	30	(3)
FILL BUNDLE INDEX	FILL AREA INDEX	37	
INTERIOR STYLE	FILL AREA INTERIOR STYLE	38	
FILL COLOUR	FILL AREA COLOUR INDEX	40	(2)
HATCH INDEX	FILL AREA STYLE INDEX	39	
PATTERN INDEX	FILL AREA STYLE INDEX	39	
PATTERN SIZE	PATTERN SIZE	41	

FILL REFERENCE POINT	PATTERN REFERENCE POINT	42	
ASPECT SOURCE FLAGS	ASPECT SOURCE FLAGS	43	(5)
PICK IDENTIFIER	PICK IDENTIFIER	50	

NOTES:

- 1) The default specification modes 'scaled' applies.
- 2) The default colour selection mode 'indexed' applies.
- 3) Four CGM elements supply the relevant parameter values of the GKS TEXT FONT AND PRECISION item (either explicitly or implicitly by default): TEXT FONT INDEX, TEXT PRECISION, CHARACTER SET INDEX and ALTERNATE CHARACTER SET INDEX. The corresponding GKS font number may be determined as described in sub-clause G.4.8.2. The occurrence of only one of the four CGM elements uniquely indicates the mapping to GKS TEXT FONT AND PRECISION. The occurrence of more than one CGM element within one sequence in any order causes the corresponding GKS item to be returned once.
- 4) Two CGM elements supply the relevant parameter values of the GKS CHARACTER VECTORS item (either explicitly or implicitly by default): CHARACTER HEIGHT and CHARACTER ORIENTATION. The occurrence of only one of the two CGM elements uniquely indicates the mapping to GKS CHARACTER VECTORS. The occurrence of the two CGM elements within one sequence in any order causes the corresponding GKS item to be returned once.
- 5) TEXT FONT ASF and TEXT PRECISION ASF must be equal: they correspond to GKS TEXT FONT AND PRECISION ASF. HATCH and PATTERN INDEX ASF must be equal: they correspond to GKS FILL ARF & STYLE INDEX ASF.

G.6.8 Mapping of escape and external elements

CGM Element	GKS Metafile Interface	Item	Notes
ESCAPE	ESCAPE	6	
MESSAGE	MESSAGE	5	(1)
APPLICATION DATA	USER ITEM	>100	

NOTES

- 1) The 'action required' flag should be 'no_action'.

G.6.9 Mapping of segment control elements

CGM Element	GKS Metafile Interface	Item	Notes
COPY SEGMENT	-	-	
INHERITANCE FILTER	-	-	

G.6.10 Mapping of segment attribute elements

CGM Element	GKS Metafile Interface	Item	Notes
SEGMENT TRANSFORMATION	SET SEGMENT TRANSFORMATION	91	
SEGMENT HIGHLIGHTING	SET HIGHLIGHTING	93	
SEGMENT DISPLAY PRIORITY	SET SEGMENT PRIORITY	94	(1)
SEGMENT PICK PRIORITY	SET SEGMENT PRIORITY	94	(1)

NOTES:

- 1) Both CGM SEGMENT DISPLAY PRIORITY and SEGMENT PICK PRIORITY supply the parameter value of the GKS SET SEGMENT PRIORITY item.

CGM Addendum 1

Part 2

Draft Copy

August 1989

Add the following to the end of sub-clause 5.3:

3/8 for Segment Control Elements and Segment Attribute Elements
3/13 for Raster Elements

Add the following to table 1:

opcode	7bit coding		8 bit coding	
BEGIN SEGMENT opcode	3/0	2/5	03/0	02/5
END SEGMENT opcode	3/0	2/6	03/0	02/6
NAME PRECISION opcode	3/1	3/0	03/1	03/0
MAXIMUM VDC EXTENT opcode	3/1	3/1	03/1	03/1
SEGMENT PRIORITY EXTENT opcode	3/1	3/2	03/1	03/2
DEVICE VIEWPORT opcode	3/2	2/7	03/2	02/7
DEVICE V'PORT SPEC. MODE opcode	3/2	2/8	03/2	02/8
DEVICE VIEWPORT MAPPING opcode	3/2	2/9	03/2	02/9
LINE REPRESENTATION opcode	3/2	2/10	03/2	02/10
MARKER REPRESENTATION opcode	3/2	2/11	03/2	02/11
TEXT REPRESENTATION opcode	3/2	2/12	03/2	02/12
FILL REPRESENTATION opcode	3/2	2/13	03/2	02/13
EDGE REPRESENTATION opcode	3/2	2/14	03/2	02/14
LINE CLIPPING MODE	3/3	2/6	03/3	02/6
MARKER CLIPPING MODE	3/3	2/7	03/3	02/7
EDGE CLIPPING MODE	3/3	2/8	03/3	02/8
BEGIN FIGURE opcode	3/3	2/9	03/3	02/9
END FIGURE opcode	3/3	2/10	03/3	02/10
NEW REGION opcode	3/3	2/11	03/3	02/11
SAVE PRIMITIVE CONTEXT opcode	3/3	2/12	03/3	02/12
RESTORE PRIMITIVE CONTEXT op.	3/3	2/13	03/3	02/13
CIRCULAR ARC CENTRE REV'D op.	3/4	2/8	03/4	02/8
CONNECTING EDGE opcode	3/4	2/9	03/4	02/9
PICK IDENTIFIER opcode	3/6	3/2	03/6	03/2
COPY SEGMENT opcode	3/8	2/0	03/8	02/0
INHERITANCE FILTER opcode	3/8	2/1	03/8	02/1
CLIP INHERITANCE opcode	3/8	2/2	03/8	02/2
SEGMENT TRANSFORMATION opcode	3/8	2/3	03/8	02/3
SEGMENT HIGHLIGHTING opcode	3/8	2/4	03/8	02/4
SEGMENT DISPLAY PRIORITY opcode	3/8	2/5	03/8	02/5
SEGMENT PICK PRIORITY opcode	3/8	2/6	03/8	02/6

Add the following sub-clause after sub-clause 6.12:

6.13 Viewport Point parameters

A viewport Point (VP) is a pair of VC scalars (Viewport Coordinate) representing the x and y coordinates of a point in viewport specification space. A VC scalar is either an integer or real number according to whether VIEWPORT SPECIFICATION MODE is 'fraction of display surface', 'millimetres with scale factor' or 'physical device units'.

When VIEWPORT SPECIFICATION MODE is 'fraction of display surface', the encoding of the viewport point data type is as described in 6.4 Coding Real Numbers. The size of the viewport point parameters is limited by the current REAL PRECISION value.

When VIEWPORT SPECIFICATION MODE is 'millimetres with scale factor' or 'physical device units', the encoding of the viewport point data type is as described in 6.3 Coding Integers. The size of the viewport point parameters is limited by the current INTEGER PRECISION value.

6.14 Name parameters

Name parameters are coded as integers (Basic format) at NAME PRECISION.

Page 31

The following form sub-clauses 8.1.6 and 8.1.7

8.1.6 BEGIN SEGMENT

<BEGIN SEGMENT opcode: 3/0 2/5>
<name: segment-identifier>

8.1.7 END SEGMENT

<END SEGMENT opcode: 3/0 2/6>

Page 34

Add the following to the <enumerated: element set> of sub-clause 8.2.11:

<integer:2>	(VER.2 STATIC SET)
<integer:3>	(EXTENDED PRIMITIVES SET)
<integer:4>	(VER.2 STATIC GKSM SET)

Page 36

The following form sub-clauses 8.2.16 to 8.2.18

8.2.16 NAME PRECISION

<NAME PRECISION opcode: 3/1 3/0>
<integer: largest-name-code + 1>

NOTE: The largest-name-code indicates how many bits occur in the largest possible magnitude for a name.

8.2.17 MAXIMUM VDC EXTENT

<MAXIMUM VDC EXTENT opcode: 3/1 3/1>
<point: first corner>
<point: second corner>

8.2.18 SEGMENT PRIORITY EXTENT

<SEGMENT PRIORITY EXTENT opcode: 3/1 3/2>
<integer: minimum-segment-priority-value>
<integer: maximum-segment-priority-value>

Page 38

The following form sub-clauses 8.3.8 to 8.3.15

8.3.8 DEVICE VIEWPORT

<DEVICE VIEWPORT opcode: 3/2 2/7>
<vp: first corner>)
<vp: second corner>)

8.3.9 DEVICE VIEWPORT SPECIFICATION MODE

<DEVICE VIEWPORT SPECIFICATION MODE op-code: 3/2 2/8>

<enumerated: VC specifier>

<real: metric scale factor>

<enumerated: VC specifier>

- <integer: 0> (fraction of drawing surface)
- ! <integer: 1> (mm with scale factor)
- ! <integer: 2> (physical device units)

8.3.10 DEVICE VIEWPORT MAPPING

<DEVICE VIEWPORT MAPPING op-code: 3/2 2/9>

<enumerated: isotropy-flag>

<enumerated: horizontal-alignment-flag>

<enumerated: vertical-alignment-flag>

<enumerated: isotropy-flag>

- <integer: 0> (not forced)
- ! <integer: 1> (forced)

<enumerated: horizontal-alignment-flag>

- <integer: 0> (left)
- ! <integer: 1> (centre)
- ! <integer: 2> (right)

<enumerated: vertical-alignment-flag>

- <integer: 0> (bottom)
- ! <integer: 1> (centre)
- ! <integer: 2> (top)

8.3.11 LINE REPRESENTATION

<LINE REPRESENTATION op-code: 3/2 2/10>

<index: line-bundle-index>

<index: line-type>

<line-width-specifier>

<colour-specifier>

<index: line-bundle-index>

- <positive integer>
- <integer: 1> (solid)
- ! <integer: 2> (dash)
- ! <integer: 3> (dot)
- ! <integer: 4> (dash-dot)
- ! <integer: 5> (dash-dot-dot)
- ! <integer: negative> (private line type)

<index: line-type>

<line-width-specifier>

- <real: line width-scale-factor>
(if LINE WIDTH SPECIFICATION MODE is scaled)
- ! <VDC: line width>
(if LINE WIDTH SPECIFIC MODE IS absolute)

<colour-specifier>

- <integer: colour index>
(if COLOUR SELECTION MODE is indexed)
- ! <RGB>
(if COLOUR SELECTION MODE IS direct)

<integer: colour-index>

- <non-negative integer>

8.3.12 MARKER REPRESENTATION

<MARKER-REPRESENTATION op-code: 3/2 2/11>

<index: marker-bundle-index>

<index: marker-type>

<marker-size-specifier>

<colour-specifier>

<index: marker-bundle-index>

- <positive integer>
- <integer: 1> (dot)
- ! <integer: 2> (plus)
- ! <integer: 3> (asterisk)
- ! <integer: 4> (circle)
- ! <integer: 5> (cross)

<index: marker-type>

<marker-size-specifier>

<colour-specifier>

<integer: colour-index>

| <integer: negative> (private marker type)

■ <real: marker-size-scale-factor>

(if MARKER SIZE SPEC MODE is scaled)

| <VDC: marker-size>

(if MARKER-SIZE SPEC MODE is absolute)

■ <integer: colour-index>

(if COLOUR SELECTION MODE is indexed)

| <RGB>

(if COLOUR SELEC MODE is direct)

■ <non-negative integer>

8.3.13 TEXT REPRESENTATION

<TEXT-REPRESENTATION-opcode: 3/2 2/2>

<index: text-bundle-index>

<integer: text-font-index>

<enumerated: text-precision>

<real: character-spacing>

<real: expansion-factor>

<colour-specifier>

<index: text-bundle-index>

<integer: text-font-index>

<enumerated: text-precision>

<real: expansion-factor>

■ <positive integer>

■ <positive integer>

■ <integer:0> (string)

| <integer:1> (character)

| <integer:2> (stroke)

■ <non-negative real>

8.3.14 FILL REPRESENTATION

<FILL-REPRESENTATION-opcode: 3/2 2/3>

<index: fill-bundle-index>

<enumerated: interior-style>

<colour-specifier>

<index: hatch-index>

<index: pattern-index>

<index: fill-bundle-index>

<enumerated: interior style>

■ <positive integer>

■ <integer:0> (hollow)

| <integer:1> (solid)

| <integer:2> (pattern)

| <integer:3> (hatch)

| <integer:4> (empty)

| <integer:negative> (private style)

<colour-specifier>

■ <integer: colour-index>

(if COLOUR SELECTION MODE is indexed)

| <RGB>

(if COLOUR SELECTION MODE is direct)

■ <integer:1> (horizontal)

| <integer:2> (vertical)

| <integer:3> (positive slope)

| <integer:4> (negative slope)

| <integer:5> (horiz/vertical cross)

| <integer:6> (positive/neg cross)

| <integer:negative> (private styles)

<index: pattern-index>

■ <positive integer>

8.3.15 EDGE REPRESENTATION

<EDGE-REPRESENTATION-opcode: 3/2 2/4>

<index: edge-bundle-index>

<index: edge-type>
 <edge-width-specifier>
 <colour-specifier>

<end: edge-bundle-index>
 <index: edge-type>

<edge-width-specifier>

<colour-specifier>

<integer: colour-index>

- <positive integer>
- <integer: 1> (solid)
- <integer: 2> (dash)
- <integer: 3> (dot)
- <integer: 4> (dash-dot)
- <integer: 5> (dash-dot-dot)
- <integer: negative> (private edge type)
- <real: edge-width-scale-factor>
 (if EDGE WIDTH SPEC MODE is scaled)
- <VDC: edge width>
 (if EDGE WIDTH SPEC MODE is absolute)
- <integer: colour-index>
 (if COLOUR SELECTION MODE is indexed)
- <RGB>
 (if COLOUR SELECTION MODE is direct)
- <non-negative integer>

Page 40

The following form sub-clauses 8.4.7 to 8.4.15

8.4.7 LINE CLIPPING MODE

<LINE CLIPPING MODE-opcode>: 3/3 2/6>
 <enumerated: clipping mode>

<enumerated: clipping mode>

- <integer:0> (locus)
- <integer:1> (shape)
- <integer:2> (locus then shape)

8.4.8 MARKER CLIPPING MODE

<LINE CLIPPING MODE-opcode>: 3/3 2/7>
 <enumerated: clipping mode>

<enumerated: clipping mode>

- <integer:0> (locus)
- <integer:1> (shape)
- <integer:2> (locus then shape)

8.4.9 EDGE CLIPPING MODE

<LINE CLIPPING MODE-opcode>: 3/3 2/8>
 <enumerated: clipping mode>

<enumerated: clipping mode>

- <integer:0> (locus)
- <integer:1> (shape)
- <integer:2> (locus then shape)

8.4.10 BEGIN FIGURE

<BEGIN FIGURE-opcode 3/3 2/9>

8.4.11 END FIGURE

<END FIGURE-opcode: 3/3 2/10>

8.4.12 NEW REGION

<NEW REGION-opcode: 3/3 2/11>

8.4.13 SAVE PRIMITIVE CONTEXT

<SAVE PRIMITIVE ATTRIBUTES-opcode: 3/3 2/12>
<name: context>

8.4.14 RESTORE PRIMITIVE CONTEXT

<RESTORE PRIMITIVE ATTRIBUTES-opcode: 3/3 2/13>
<name: context>

Page 45

The following form sub-clause 8.5.20 and 8.5.21

8.5.20 CIRCULAR ARC CENTRE REVERSED

<CIRCULAR ARC CENTRE REVERSED-opcode: 3/4 2/8>
<point: centrepoint>
<VDC: DX_start>
<VDC: DY_start>
<VDC: DX_end>
<VDC: DY_end>
<VDC: radius>

8.5.21 CONNECTING EDGE

<CONNECTING EDGE-opcode: 3/4 2/9>

Page 54

The following forms sub-clause 8.6.36

8.6.36 PICK IDENTIFIER

<PICK-ID-opcode: 3/6 3/2>
<name: pick-identifier>

Page 56

The following forms sub-clause 8.9

8.9.1 COPY SEGMENT

<COPY-SEGMENT-opcode: 3/8 2/0>
<name: segment-identifier>
<transformation matrix>
<source: segment transformation application>

<transformation matrix> = <real: a11>
<real: a12>
<real: a21>
<real: a22>
<vdc: a13>
<vdc: a23>

<source: segment transformation application> = <integer: 0> (no)

8.9.2 INHERITANCE FILTER

<INHERITANCE-FILTER-opcode: 3/8 2/1>

<enumerated: filter-selection-list>

<enumerated: selection settings>

<enumerated: filter-selection-list>

<integer:1>	(yes)
<integer:0>	(line bundle index)
<integer:1>	(line type)
<integer:2>	(line width)
<integer:3>	(line colour)
<integer:4>	(line clipping mode)
<integer:5>	(marker bundle index)
<integer:6>	(marker type)
<integer:7>	(marker size)
<integer:8>	(marker colour)
<integer:9>	(marker clipping mode)
<integer:10>	(text bundle index)
<integer:11>	(text font index)
<integer:12>	(text precision)
<integer:13>	(character expansion factor)
<integer:14>	(character spacing)
<integer:15>	(text colour)
<integer:16>	(character height)
<integer:17>	(character orientation)
<integer:18>	(text path)
<integer:19>	(text alignment)
<integer:22>	(fill bundle index)
<integer:23>	(interior style)
<integer:24>	(fill colour)
<integer:25>	(hatch index)
<integer:26>	(pattern index)
<integer:27>	(edge bundle index)
<integer:28>	(edge type)
<integer:29>	(edge width)
<integer:30>	(edge colour)
<integer:31>	(edge visibility)
<integer:32>	(edge clipping mode)
<integer:33>	(fill reference point)
<integer:34>	(pattern size)
<integer:35>	(auxiliary colour)
<integer:36>	(transparency)
<integer:38>	(line attributes)
<integer:39>	(marker attributes)
<integer:40>	(text attributes)
<integer:41>	(character attributes)
<integer:42>	(fill attributes)
<integer:43>	(edge attributes)
<integer:44>	(pattern attributes)
<integer>	(output control)
<integer>	(pick identifier)
<integer:45>	(all attributes and control)
<integer>	(all)
<integer>	(line type ASF)
<integer>	(line width ASF)
<integer>	(line colour ASF)
<integer>	(marker type ASF)
<integer>	(marker size ASF)
<integer>	(marker colour ASF)
<integer>	(text font index ASF)
<integer>	(text PRECISION ASF)
<integer>	(character expansion factor)

<integer>	(character spacing ASF)
<integer>	(text colour ASF)
<integer>	(interior style ASF)
<integer>	(fill colour ASF)
<integer>	(hatch index ASF)
<integer>	(pattern index ASF)
<integer>	(edge type ASF)
<integer>	(edge width ASF)
<integer>	(edge colour ASF)
<integer:46>	(line ASFs)
<integer:47>	(marker ASFs)
<integer:48>	(text ASFs)
<integer:49>	(character ASFs)
<integer:50>	(fill ASFs)
<integer:51>	(edge ASFs)
<integer:52>	(all ASFs)

<enumerated: selection setting>	=	<integer:0>	(state_list)
		<integer:1>	(segment)

8.9.3 CLIP INHERITANCE

<CLIP INHERITANCE-opcode: 3/8 2/2>
 <name: clip-inheritance>
 <enumerated: setting>

=	<integer:0>	(state_list)
	<integer:1>	(intersection)

8.9.4 SEGMENT TRANSFORMATION

<SEGMENT-TRANSFORMATION-opcode: 3/8 2/3>
 <name: segment-identifier>
 <transformation matrix>

<transformation matrix>

=	<real: a11 >
	<real: a12 >
	<real: a21 >
	<real: a22 >
	<real: a13 >
	<real: a23 >

8.9.5 SEGMENT HIGHLIGHTING

<SEGMENT-HIGHLIGHTING-opcode: 3/8 2/4>
 <name: segment-identifier>
 <enumerated: segment-highlighting>

<name: segment identifier>

=	<integer>	
=	<integer: 0>	(normal)
	<integer: 1>	(highlighted)

8.9.6 SEGMENT DISPLAY PRIORITY

<SEGMENT-PRIORITY-opcode: 3/8 2/5>
 <name: segment-identifier>
 <integer: segment-display-priority>

<integer: segment-display-priority>

=	<positive integer>
---	--------------------

8.9.7 SEGMENT PICK PRIORITY

<SEGMENT-PICK-PRIORITY-opcode: 3/8 2/6>
 <name: segment-identifier>

<integer: pick:priority>

<integer: pick:priority> = <positive integer>

Page 56

Add the following to the end of clause 9

NAME PRECISION : 10

Page 60

Add the following to the end of Annex A

<name precision value> = <integer> (see 8.2)

<viewport point> = <integer><integer>

! <real><real>

<name> = <integer>

<2x2 matrix of reals> = <real>(4) (see 8.9)

<2x1 matrix of vdc's> = <vdc value>(2) (see 8.9)

CGM Addendum 1

Part 3

Draft Copy

August 1989

Page 16

Add the following to table 1:

N	SI as integer precision (np)	BN (=np/8)	NR (-2 ^{np} (np-1) to 2 ^{np} (np-1)-1)
VP	(R,R) or (L)	BVSP (=2*BR) BVSP (=2*BI)	VSPR = (RR) VSPR = (IR)

Page 19

Add the following to Table 2:

8 Segment Control and Segment Attribute elements

Page 20

Add the following to table 3:

BEGIN SEGMENT	6	N	BN	NR	n/a
END SEGMENT	7	n/a	0	n/a	n/a

Code Description

- 6 BEGIN SEGMENT: has 1 parameter:
P1: (segment name) segment identifier
- 7 END SEGMENT: has no parameters

Page 21

Add the following to table 4:

NAME PRECISION	16	N	BN	8,16,24,32	16
MAXIMUM VDC EXTENT	17	2P	2BP	VDCR	VDC EXTENT
SEG PRIORITY EXTENT	18	2I	2BI	IR	0, 255

Code Description

- 16 NAME PRECISION: has 1 parameter:
P1: (integer) name precision: 8,16,24 or 32 are the only valid values
- 17 MAXIMUM VDC EXTENT: has 2 parameters:
P1: (point) first point
P2: (point) second point
- 18 SEGMENT PRIORITY EXTENT: has 2 parameters:
P1: (integer) minimum segment priority value
P2: (integer) maximum segment priority value

Add to the note P2 of METAFILE ELEMENT LIST:

ver.2-static (-1,2)
 extended-primitives (-1,3)
 ver.2-gism-static (-1,4)

Add to the end of the note P2 for SCALING MODE:

Note that this parameter is always encoded as Floating Point, regardless of the value of the fixed/floating flag of REAL PRECISION.

Add the following to table 5:

DEVICE VIEWPORT	8	2VP	2BVP	VPR	see below
DEVICE VIEWPORT SPECIFICATION MODE	9	E.R	BE+BR	(0,1,2)	0,-
DEVICE VIEWPORT MAPPING	10	3E	3BE	(0,1) (0,1,2) (0,1,2)	1 0 0
LINE REPRESENTATION	11	2IX, (VDC or R),CO	2BIX+ (BVDC or BR)+BCO	+IXR,IXR, ↔VDCR or ↔RR,COR	n/a
MARKER REPTION	12	2IX, (VDC or R),CO	2BIX+ (BVDC or BP)+BCO	+IXR,IXR, ↔VDCR or ↔RR,COR	n/a
TEXT REPRESENTATION	13	2IX, E, 2R.CO	2BIX+ BE+ 2BR+BCO	+IXR, (0,1,2), RR,COR	n/a
FILL REPRESENTATION	14	IX, E.CO,	BIX+ BE+BCO+	+IXR, (0,4),COR,	n/a
EDGE REPRESENTATION	15	2IX, (VDC or R),CO	2BIX+ (BVDC or BR)+BCO	+IXR,IXR, ↔VDCR or ↔RR,COR	n/a

Code Description

8 DEVICE VIEWPORT: has 2 parameters:

P1: (vp) first point
 P2: (vp) second point

The default DEVICE VIEWPORT is the entire device view surface if the latter is rectangular, or the largest rectangular subset having the desired aspect ratio, if the view surface is not rectangular. The default is set so that the 'first point' is below and to the left of the 'second point' as seen by the viewer.

9 DEVICE VIEWPORT SPECIFICATION MODE: has 2 parameters:

P1: (enumerated) viewport specification units: valid values are:

- 0 fraction of drawing surface
- 1 millimetres with scale factor
- 2 physical device units

P2: (real) metric scale factor, ignored if P1=0 or P1=2

Note that this parameter is always encoded as Floating Point, regardless of the value of the fixed/floating flag of REAL PRECISION.

- 10 **DEVICE VIEWPORT MAPPING:** has 3 parameters:
- P1: (enumerated) isotropy flag: valid values are:
- 0 not forced
- 1 forced
- P2: (enumerated) horizontal alignment flag: valid values are:
- 0 left
- 1 centre
- 2 right
- P3: (enumerated) vertical alignment flag: valid values are:
- 0 bottom
- 1 centre
- 2 top
- 11 **LINE REPRESENTATION:** has 4 parameters:
- P1: (index) line bundle index
- P2: (index) line type: the following values are standardized:
- 1 solid
- 2 dash
- 3 dot
- 4 dash-dot
- 5 dash-dot-dot
- negative for private use
- P3: (vdc or real) absolute line width or line width scale factor
- P4: (colour) line colour: its form depends on COLOUR SELECTION MODE.
- 12 **MARKER REPRESENTATION:** has 4 parameters:
- P1: (index) marker bundle index
- P2: (index) marker type: the following values are standardized:
- 1 dot
- 2 plus
- 3 asterisk
- 4 circle
- 5 cross
- negative for private use
- P3: (vdc or real) absolute marker width or marker size scale factor
- P4: (colour) marker colour: its form depends on COLOUR SELECTION MODE.
- 13 **TEXT REPRESENTATION:** has 6 parameters:
- P1: (index) text bundle index
- P2: (index) text font index
- P3: (enumerated) text precision: valid values are:
- 0 string
- 1 character
- 2 stroke
- P4: (real) character spacing
- P5: (real) character expansion factor
- P6: (colour) text colour: its form depends on COLOUR SELECTION MODE
- 14 **FILL REPRESENTATION:** has 5 parameters:
- P1: (index) fill area bundle index
- P2: (enumerated) interior style: valid values are:
- 0 hollow
- 1 solid
- 2 pattern
- 3 hatch
- 4 empty
- P3: (colour) fill colour: its form depends on COLOUR SELECTION MODE
- P4: (index) hatch index: the following values are standardized:
- 1 horizontal

4 vertical
 3 positive slope
 4 negative slope
 5 combined vertical and horizontal slant
 6 combined left and right slant
 negative for private use
 P5: (index) pattern index

15 **EDGE REPRESENTATION:** has 4 parameters:

P1: (index) edge bundle index
 P2: (index) edge type: the following values are standardized:
 1 solid
 2 dash
 3 dot
 4 dash-dot
 5 dash-dot-dot
 negative for private use
 P3: (vdc or real) absolute edge width or line width scale factor
 P4: (colour) edge colour: its form depends on COLOUR SELECTION MODE

Page 26

Add the following to table 6:

LINE CLIPPING MODE	7	E	EE	(0,1,2)	0
MARKER CLIPPING MODE	8	E	EE	(0,1,2)	0
EDGE CLIPPING MODE	9	E	EE	(0,1,2)	0
BEGIN FIGURE	10	n/a	0	n/a	n/a
END FIGURE	11	n/a	0	n/a	n/a
NEW REGION	12	n/a	0	n/a	n/a
SAVE PRIMITIVE CONTEXT	13	N	BN	NR	n/a
RESTORE PRIMITIVE CONTEXT	14	N	BN	BN	NR

7 **LINE CLIPPING MODE:** has 1 parameter:

P1: (enumerated) clipping mode: valid values are:
 0 locus
 1 shape
 2 locus then shape

8 **MARKER CLIPPING MODE:** has 1 parameter:

P1: (enumerated) clipping mode: valid values are:
 0 locus
 1 shape
 2 locus then shape

9 **EDGE CLIPPING MODE:** has 1 parameter:

P1: (enumerated) clipping mode: valid values are:
 0 locus
 1 shape
 2 locus then shape

10 **BEGIN FIGURE:** has no parameters

11 **END FIGURE:** has no parameters

12 **NEW REGION:** has no parameters

13 **SAVE PRIMITIVE CONTEXT:** has 1 parameter:

P1: (name) context name

14 RESTORE PRIMITIVE CONTEXT: has 1 parameter:

P1: (name) context name

Page 28

Add the following to table 7:

CIRCULAR ARC CENTRE REVERSED	20	P,4VDC, VDC	BP+4BVDC+ BVDC	VDCR,VDCR, +VDCR	n/a
CONNECTING EDGE	21	n/a	0	n/a	n/a

Code Description

20 CIRCULAR ARC CENTRE REVERSED: has 6 parameters:

P1: (point) centre of circle
P2: (vdc) delta X for start vector
P3: (vdc) delta Y for start vector
P4: (vdc) delta X for end vector
P5: (vdc) delta Y for end vector
P6: (vdc) radius of circle

21 CONNECTING EDGE has no parameters

Page 33

Add the following to table 8:

PICK IDENTIFIER	36	N	BN	NR	0
-----------------	----	---	----	----	---

Code Description

36 PICK IDENTIFIER: has 1 parameter:

P1: (name) pick identifier

Page 39

The following forms sub-clause 7.10

7.10 Segment Control and Segment Attribute Elements

Table 11 - Encoding of segment control and segment attribute elements

Element Class #	EI Id	Param Type	Parameter List Length	Parameter Range	Default
COPY SEGMENT	1	N,4R, 2VDC, E	BN+4BR+ 2BVDC + BE	NR,RR, VDCR, {0,1}	-0
INHERITANCE FILTER	2	nE,E	(n+1)BE	{0,43},{0,1}	-1
CLIP INHERITANCE	3	E	BE	{0,1}	0
SEGMENT TRANSFORM	4	N,4R, 2VDC	BN+4BR+ 2BVDC	NR,RR, VDCR	n/a,1,0,0,0, 1,0
SEGMENT HIGHLIGHTING	5	N,E	BN+BE	NR,{0,1}	n/a,0
SEG DISPLAY PRIORITY	6	N,I	BN+BI	NR,IR	n/a,see below
SEGMENT PICK PRIORITY	7	N,I	BN+BI	NR,IR	n/a,see below

Code Description

1 COPY SEGMENT: has 8 parameters:

P1: (name) segment identifier

The next 6 parameters are components of a 3x2 matrix of the form...

(P2 P3 P6)
(P4 P5 P7)

where:

P2: (real) x scale component

P3: (real) x rotation component

P4: (real) y rotation component

P5: (real) y scale component

P6: (vdc) x translation component

P7: (vdc) y translation component

P8: (enumerated) segment transformation application: valid values are:

0: no

1: yes

- 2 INHERITANCE FILTER: has two parameters. The first is a list of up to attribute or group designators. The second is a single setting value.

P1: (enumerated list) list of one or more of:

- 0 line bundle index
- 1 line type
- 2 line width
- 3 line colour
- line clipping mode
- 4 marker bundle index
- 5 marker type
- 6 marker size
- 7 marker colour
- marker clipping mode
- 8 text bundle index
- 9 text font index
- 10 text precision
- 11 character expansion factor
- 12 character spacing
- 13 text colour
- 14 character height
- 15 character orientation
- 16 text path
- 17 text alignment
- 20 fill bundle index
- 21 interior style
- 22 fill colour
- 23 hatch index
- 24 pattern index
- 25 edge bundle index
- 26 edge type
- 27 edge width
- 28 edge colour
- 29 edge visibility
- edge clipping mode
- 30 fill reference point
- 31 pattern size
- 32 auxiliary colour
- 33 transparency
- 34 line attributes
- 35 marker attributes
- 36 text attributes
- 37 character attributes
- 38 fill attributes
- 39 edge attributes
- 40 pattern attributes
- 42 output control
- pick identifier
- 43 all attributes and control
- all

line type asf
 line width asf
 line colour asf
 marker type asf
 marker size asf
 marker colour asf
 text font index asf
 text precision asf
 character expansion factor asf
 character spacing asf
 text colour asf
 interior style asf
 fill colour asf
 hatch index asf
 pattern index asf
 edge type asf
 edge width asf
 edge colour asf
 line asfs
 marker asfs
 text asfs
 fill asfs
 edge asfs
 all asfs

P2: (enumerated) setting: valid values are:

0 state_list
 1 segment

3 CLIP INHERITANCE: has 1 parameter

P1: (enumerated) clip inheritance: valid values are:

0 state_list
 1 segment

4 SEGMENT TRANSFORMATION: has 2 parameters:

P1: (name) segment identifier

The second parameter consists of a 3x2 matrix of the form:

P2: (2x2 matrix of reals, 2x1 matrix of vdc's)

5 SEGMENT HIGHLIGHTING: has 2 parameters:

P1: (name) segment identifier

P2: (enumerated) highlighting: valid values are:

0 normal
 1 highlighted

6 SEGMENT DISPLAY PRIORITY: has 2 parameters:

P1: (name) segment identifier

P2: (integer) segment display priority

The default of the segment display priority is equal to the minimum segment priority value (see sub-clause 7.3)

7 SEGMENT PICK PRIORITY: has 2 parameters:

P1: (name) segment identifier

P2: (integer) segment pick priority

The default of the segment pick priority is equal to the minimum segment priority value (see sub-clause 7.3)

Page 40

Add the following to the end of clause 8:

NAME PRECISION 16

Page 42

Add the following to the list of references:

<viewport point> => <integer(2)/real(2)>

<name> => <integer>

<2x2 matrix of reals> => <real>(4)

<2x1 matrix of vdc's> => <vdc value>(2)

Page 48

Add the following to the list of elements:

Class	Element Code	Element Name
0	6	BEGIN SEGMENT
0	7	END SEGMENT
1	16	NAME PRECISION
1	17	MAXIMUM VDC EXTENT
1	18	SEGMENT PRIORITY EXTENT
2	8	DEVICE VIEWPORT
2	9	DEVICE VIEWPORT SPECIFICATION MODE
2	10	DEVICE VIEWPORT MAPPING
2	11	LINE REPRESENTATION
2	12	MARKER REPRESENTATION
2	13	TEXT REPRESENTATION
2	14	FILL REPRESENTATION
2	15	EDGE REPRESENTATION
3	7	LINE CLIPPING MODE
3	8	MARKER CLIPPING MODE
3	9	EDGE CLIPPING MODE
3	10	BEGIN FIGURE
3	11	END FIGURE
3	12	NEW REGION
3	13	SAVE PRIMITIVE CONTEXT
3	14	RESTORE PRIMITIVE CONTEXT
4	20	CIRCULAR ARC CENTRE REVERSED
4	21	CONNECTING EDGE
5	36	PICK IDENTIFIER
8	1	COPY SEGMENT
8	2	INHERITANCE FILTER
8	3	CLIP INHERITANCE
8	4	SEGMENT TRANSFORMATION
8	5	SEGMENT HIGHLIGHTING
8	6	SEGMENT DISPLAY PRIORITY
8	7	SEGMENT PICK PRIORITY

CGM Addendum i

Part 4

Draft Copy

August 1989

Add the following to the end of sub-clause 5.3.5

N := <I> (name)

VC := <R><I> (viewport coordinate data depending
 on DEVICE VIEWPORT SPECIFICATION MODE)

VPOINTREC := <VC><SEP><VC>

VP := <VPOINTREC><LEFT PAREN><OPTSEP><VPOINTREC><OPTSEP>
 <RIGHT PAREN>

(COORDINATE in viewport specification space. Parentheses are optional. If they are used, they shall group exactly two real or integer numbers, depending on DEVICE VIEWPORT SPECIFICATION MODE. The parenthesized form is intended to aid readability of the metafile.)

ROWMATRIX := <<R:FIRST ELEMENT IN ROW>
 <SEP>
 <R:SECOND ELEMENT IN ROW>
 <SEP>
 <VDC:LAST ELEMENT IN ROW>>>
 <<LEFT PAREN>
 <OPTSEP>
 <R:FIRST ELEMENT IN ROW>
 <SEP>
 <R:SECOND ELEMENT IN ROW>
 <SEP>
 <VDC:LAST ELEMENT IN ROW>
 <OPTSEP>
 <RIGHT PAREN>>>

TM := <ROWMATRIX>
 <SEP>
 <ROWMATRIX>

(2x3 transformation matrix in row-major order. Parentheses are optional. If they are used, they shall group exactly two real numbers and one VDC number. The parenthesized form is intended to aid readability of the metafile. If there are not three numbers in each parenthesized group, the metafile is non-conforming interchange. Any attempted error recovery or exception handling which a metafile interpreter may use in this situation is neither defined nor constrained by ISO 8632/4. A metafile interpreter need not use the parentheses in parsing; in this case, they are treated as SPACE characters rather than as NULL characters (i.e., they act as soft delimiters).)

Add the following to the end of sub-clause 5.4.3

ALL
COPY
EXTENDED
FIGURE
FILTER
FORCED
FRACTION
GKSM
INTERSECTION
LOCUS
MATRIX
NAME
NEW
PICK
PRIMITIVES
REGION

SAVE
SHAPE
STATIC
THEN
UNITS
VER.2

Page 12

Add the following to the end of sub-clause 5.4.4

ATTRIBUTE(S)	ATTR
CLIPPING	CLIP
CONNECTING	CONN
CONTEXT	CONT
DEVICE	DEV
DISPLAY	DISP
HIGHLIGHTING	HIGHLIGHT
IDENTIFIER	ID
INHERITANCE	INH
MAPPING	MAP
MILLIMETER	MM
PHYSICAL	PHY
PIXEL	PEL
PRIMITIVE	PRIM
PRIORITY	PRI
REPRESENTATION	REP
RESTORE	RES
REVERSED	REV
SEGMENT	SEG
TRANSFORMATION	TRAN
VIEWPORT	VP

Page 14

Add the following to the end of sub-clause 5.4.5

BEGIN SEGMENT	BEGSEG
END SEGMENT	ENDSEG
NAME PRECISION	NAMEPREC
MAXIMUM VDC EXTENT	MAXVDCEXT
SEGMENT PRIORITY EXTENT	SEGPRIEXT
DEVICE VIEWPORT	DEVVP
DEVICE VIEWPORT SPECIFICATION MODE	DEVVPMODE
DEVICE VIEWPORT MAPPING	DEVVPMAP
LINE REPRESENTATION	LINEREP
MARKER REPRESENTATION	MARKERREP
TEXT REPRESENTATION	TEXTREP
FILL REPRESENTATION	FILLREP
EDGE REPRESENTATION	EDGEREP
LINE CLIPPING MODE	LINECLIPMODE
MARKER CLIPPING MODE	MARKERCLIPMODE
EDGE CLIPPING MODE	EDGECLIPMODE
BEGIN FIGURE	BEGFIGURE
END FIGURE	ENDFIGURE
NEW REGION	NEWREGION
SAVE PRIMITIVE CONTEXT	SAVEPRIMCONT
RESTORE PRIMITIVE CONTEXT	RESPRIMCONT
CIRCULAR ARC CENTRE REVERSED	ARCCTRREV
CONNECTING EDGE	CONNEDGE
PICK IDENTIFIER	PICKID
COPY SEGMENT	COPYSEG
INHERITANCE FILTER	INHFILTER
CLIP INHERITANCE	CLIPINH

SEGMENT TRANSFORMATION
 SEGMENT VISIBILITY
 SEGMENT HIGHLIGHTING
 SEGMENT DISPLAY PRIORITY
 SEGMENT PICK PRIORITY

SEGTRANS
 SEGVIS
 SEGHIGHL
 SEGDISPPRI
 SEGPICKPRI

Page 15

Add the following to the end of sub-clause 6.2

BEGIN SEGMENT => BEGSEG
 <SOFTSEP>
 <N:SEGID>
 <TERM>
 END SEGMENT => ENDSEG <TERM>

Page 17

Add to the end of METAFILE ELEMENT LIST:

The words VER2STATIC, EXTENDED PRIMITIVES and VER2GKSMSTATIC may also be used in this string

Page 17

Add the following to the end of sub-clause 6.3

NAME PRECISION => NAMEPREC
 <SOFTSEP>
 <I:MININT>
 <SEP>
 <I:MAXINT>
 <TERM>
 MAX VDC EXTENT => MAXVDCEXT
 <SOFTSEP>
 <P:FIRSTCORNER>
 <SEP>
 <P:SECONDCORNER>
 <TERM>
 SEGMENT PRIORITY EXTENT => SEGPRIEXT
 <SOFTSEP>
 <I:MINSEGPRI>
 <SEP>
 <I:MAXSEGPRI>
 <TERM>

Page 18

Add the following to the end of sub-clause 6.4

DEVICE VIEWPORT => DEVVP
 <SOFTSEP>
 <VP:FIRSTCORNER>
 <SEP>
 <VP:SECONDCORNER>
 <TERM>
 DEVICE VIEWPORT SPECIFICATION
 MODE ::= DEVVP

```

        <SOFTSEP>
        <FRACTIONUMMIPHYDEVUNITS>
        <SEP>
        <R:SCALEFACTOR>
    <TERM>

DEVICE VIEWPORT MAPPING ::= DEVICEVPMAP
    <SOFTSEP>
    <NOTFORCEDIFORCED>
    <SEP>
    <LEFTICTRIRIGHT>
    <SEP>
    <BOTTOMICTRITOP>
    <TERM>

LINE REPRESENTATION => LINEREP
    <SOFTSEP>
    <I:BUNDLEINDEX> (positive)
    <SEP>
    <I:LINETYPE>
        (1=solid, 2=dash
         3=dot, 4=dash-dot
         5=dash-dot-dot
         <0 implementation dependent)
    <SEP>
    <V:LINEWIDTH> (non-negative)
    <SEP>
    <K:LINECOLR>
    <TERM>

MARKER REPRESENTATION => MARKERREP
    <SOFTSEP>
    <I:BUNDLEINDEX> (positive)
    <SEP>
    <I:MARKERTYPE>
        (1=dot, 2=plus
         3=asterisk, 4=circle
         5=cross (x)
         <0 implementation dependent)
    <SEP>
    <V:MARKERSIZE> (non-negative)
    <SEP>
    <K:MARKERCOLR>
    <TERM>

TEXT REPRESENTATION ::= TEXTREP
    <SOFTSEP>
    <I:BUNDLEINDEX> (positive)
    <SEP>
    <I:FONTINDEX> (positive)
    <SEP>
    <STRINGCHARISTROKE>
    <SEP>
    <R:SPACING>
    <SEP>
    <R:FACTOR>
    <SEP>
    <K:TEXTCOLR>
    <TERM>

FILL REPRESENTATION => FILLREP
    <SOFTSEP>
    <I:BUNDLEINDEX> (positive)
    <SEP>

```

```

<HOLLOW|SOLID|PATH|HATCH|EMPTY>
<SEP>
<K:FILLCOLR>
<SEP>
<I:HATCHINDEX>
    (1=horizontal, 2=vertical
    3=positive slope
    4=negative slope
    5=horiz/vert cross
    6=+/- slope cross
    <0 implementation dependent
<SEP>
<I:PATINDEX> (positive)
<TERM>

```

EDGE REPRESENTATION

```

=> EDGEREP
    <SOFTSEP>
    <I:BUNDLEINDEX> (positive)
    <SEP>
    <I:EDGETYPE>
        (1=solid, 2=dash
        3=dot, 4=dash-dot
        5=dash-dot-dot
        <0 implementation dependent)
    <SEP>
    <V:EDGEWIDTH> (non-negative)
    <SEP>
    <K:EDGECOLR>
<TERM>

```

Page 19

Add the following to the end of sub-clause 6.5

LINE CLIPPING MODE	=> LINECLIPMODE
	<SOFTSEP>
	<LOCUSISHAPE LOCUSTHENSHP>
	<TERM>
MARKER CLIPPING MODE	=> MARKERCLIPMODE
	<SOFTSEP>
	<LOCUSISHAPE LOCUSTHENSHP>
	<TERM>
EDGE CLIPPING MODE	=> EDGECLIPMODE
	<SOFTSEP>
	<LOCUSISHAPE LOCUSTHENSHP>
	<TERM>
BEGIN FIGURE	=> BEGFIGURE <TERM>
END FIGURE	=> ENDFIGURE <TERM>
NEW REGION	=> NEWREGION <TERM>
IMPLICIT EDGE VISIBILITY	=> IMPLDGEVIS
	<SOFTSEP>
	<OFFON>
	<TERM>
SAVE PRIMITIVE CONTEXT	=> SAVEPRIMCONT
	<SOFTSEP>

```

                                <N:CONTEXTNAME>
                                <TERM>
RESTORE PRIMITIVE CONTEXT ::= RESPRIMCONT
                                <SOFTSEP>
                                <N:CONTEXTNAME>
                                <TERM>

```

Page 24

Add the following to the end of sub-clause 6.6

```

CIRCULAR ARC CENTRE REV ::= ARCCTRREV
                                <CTRARCSPEC>
                                <TERM>
CONNECTING EDGE          ::= CONNEDG <TERM>

```

Page 28

Add the following to the end of sub-clause 6.7

```

PICK IDENTIFIER          ::= PICKID
                                <SOFTSEP>
                                <N:PICKID>
                                <TERM>

```

Page 29

The following forms sub-clause 6.10

6.10 Encoding segment control and segment attribute elements

```

COPY SEGMENT             ::= COPYSEG
                                <SOFTSEP>
                                <N:SEGID>
                                <SEP>
                                <TM:TRANSMATRIX>
                                <NOIYES>
                                <TERM>

```

*****PROPER ENCODINGS TO BE ADDED ONCE LIST AGREED *****

```

INHERITANCE FILTER       ::= INHFILTER
                                <SOFTSEP>
                                <LINEINDEX>
                                LINETYPEI
                                LINEWIDTHI
                                LINECOLRI
                                LINECLIPMODEI
                                MARKERINDEXI
                                MARKERTYPEI
                                MARKERSIZEI
                                MARKERCOLRI
                                MARKERCLIPMODEI
                                TEXTINDEXI
                                TEXTFONTINDEXI
                                TEXTPRECI
                                CHARACTEREXPAN
                                CHARACTERSPACEI
                                TEXTCOLRI

```

CHARHEIGHT
 CHARORI
 TEXTPATH
 TEXTALIGN
 FILLINDEX
 INTSTYLE
 FILLCOLR
 HATCHINDEX
 PATINDEX
 EDGEINDEX
 EDGETYPE
 EDGEWIDTH
 EDGECOLR
 EDGEVISI
 EDGECLIPMODE
 FILLREFERENCEPT
 PATSIZE
 AUXCOLR
 TRANSPARENCY
 LINEATTR
 MARKERATTR
 TEXTATTR
 CHARATTR
 FILLATTR
 EDGEATTR
 PATATTR
 OUTPUTCONTROL
 PICKID
 ALLATTR ***and control*****
 ALL

line type asf
 line width asf
 line colour asf
 marker type asf
 marker size asf
 marker colour asf
 text font index asf
 text precision asf
 character expansion factor asf
 character spacing asf
 text colour asf
 interior style asf
 fill colour asf
 hatch index asf
 pattern index asf
 edge type asf
 edge width asf
 edge colour asf

LINEASFS
 MARKERASFS
 TEXTASFS
 CHARASFS
 FILLASFS
 EDGEASFS
 ALLASFS
 <SEP>
 <STATELISTISEGMENT>

<TERM>

CLIP INHERITANCE

= <CLIPINH
 <SOFTSEP>
 <STATE_LISTINTERSECTION>
 <TERM>

```

SEGMENT TRANSFORMATION ::= SEGTRANS
                           <SOFTSEP>
                           <N:SEGID>
                           <SEP>
                           <TM:TRANSMATRIX>
                           <TERM>

SEGMENT HIGHLIGHTING    ::= SEGHIGHL
                           <SOFTSEP>
                           <N:SEGID>
                           <SEP>
                           <NORMALHIGHLIGHTED>
                           <TERM>

SEGMENT DISPLAY PRIORITY ::= SEGDISPPRI
                           <SOFTSEP>
                           <N:SEGID>
                           <SEP>
                           <J:DISPRIORITY>
                           <TERM>

SEGMENT PICK PRIORITY   ::= SEGPICKPRI
                           <SOFTSEP>
                           <N:SEGID>
                           <SEP>
                           <J:PICKPRIORITY>
                           <TERM>

```

Page 30

Add the following to the end of clause 7:

Pick identifier:

```

NAME PRECISION:
  MININT 0
  MAXINT 32767

```

NIST-114A (REV. 3-90)		U.S. DEPARTMENT OF COMMERCE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY		1. PUBLICATION OR REPORT NUMBER NISTIR 4329
BIBLIOGRAPHIC DATA SHEET		2. PERFORMING ORGANIZATION REPORT NUMBER		3. PUBLICATION DATE JULY 1990
		4. TITLE AND SUBTITLE Graphics Standards in the Computer-aided Acquisition and Logistic Support (CALS) Program, Fiscal Year 1989, Volume 1: Test Requirements Document and Extended CGM (CGEM).		
5. AUTHOR(S) Daniel R. Benigni, Editor				
6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS) U.S. DEPARTMENT OF COMMERCE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY GAITHERSBURG, MD 20899			7. CONTRACT/GRANT NUMBER	
9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP) Office of the Secretary of Defense Production and Logistics/ Systems/ CALS Room 3B322, Pentagon Washington, D.C. 20301-8000			8. TYPE OF REPORT AND PERIOD COVERED NISTIR 10/88 - 9/89	
10. SUPPLEMENTARY NOTES				
11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.) <p>The computer-aided Acquisition and Logistic Support (CALS) Program is a DoD Industry strategy to transition from paper-intensive acquisition and logistic processes to a highly automated and integrated mode of operation for the weapon systems of the 1990s. These volumes document the accomplishments of the Graphics Software Group of the National Institute of Standards and Technology (NIST) in support of computer graphics standards for CALS in FY89. They provide a progress report on continuing graphics standards efforts related to the Computer Graphics Metafile (CGM) standard. These reports are divided into two volumes: 1, Test Requirements Document and Extended CGM (CGEM); and 2, MIL-D-28003 Revisions and CGM Registration.</p> <p>Volume 1. Completion of the first milestone toward the establishment of a comprehensive CGM (and MIL-D-28003) Test Method and ultimately a Testing Service, namely the creation of a Test Requirements Document, is detailed. This Test Requirements Document will provide the basis for developing conformance tests to determine compliance both with CGM and MIL-D-28003. Progress to inject CALS requirements into the Extended CGM (CGEM) is also detailed.</p>				
12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS) application profile; CALS; CGEM; CGM; CGM registration; conformance testing; DoD; graphics; illustration data; MIL-D-28003; test requirements document				
13. AVAILABILITY <input checked="" type="checkbox"/> UNLIMITED <input type="checkbox"/> FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS). <input type="checkbox"/> ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402. <input checked="" type="checkbox"/> ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.			14. NUMBER OF PRINTED PAGES 315 15. PRICE A14	

ELECTRONIC FORM